
On the Equivalence of Random Network Distillation, Deep Ensembles, and Bayesian Inference

Moritz A. Zanger¹ Yijun Wu¹ Pascal R. van der Vaart¹ Wendelin Böhmer¹ Matthijs T. J. Spaan¹

¹Delft University of Technology, Delft, 2628 XE, The Netherlands

Abstract

Uncertainty quantification is central to safe and efficient deployments of deep learning models, yet many computationally practical methods lack lacking rigorous theoretical motivation. Random network distillation (RND) is a lightweight technique that measures novelty via prediction errors against a fixed random target. While empirically effective, it has remained unclear what uncertainties RND measures and how its estimates relate to other approaches, e.g., Bayesian inference or deep ensembles. We establish these missing theoretical connections by analyzing RND within the neural tangent kernel framework in the limit of infinite network width. Our analysis reveals two central findings in this limit: (1) The uncertainty signal from RND—its squared self-predictive error—is equivalent to the predictive variance of a deep ensemble. (2) By constructing a specific RND target function, we show that the RND error distribution can be made to mirror the centered posterior predictive distribution of Bayesian inference with wide neural networks. Based on this equivalence, we moreover devise a posterior sampling algorithm that generates i.i.d. samples from an exact Bayesian posterior predictive distribution using this modified *Bayesian RND* model. Collectively, our findings provide a unified theoretical perspective that places RND within the principled frameworks of deep ensembles and Bayesian inference, and offer new avenues for efficient yet theoretically grounded uncertainty quantification methods.

1 INTRODUCTION

Quantifying predictive uncertainty remains a cornerstone of reliable machine learning and underpins applications from

safe robotics to efficiently exploring agents and autonomous scientific discovery. Bayesian inference is widely regarded as a theoretical gold-standard to this end [Neal, 1996, Goan and Fookes, 2020] but its application to neural networks is typically intractable in practice, requiring approximations of simplified posteriors through variational inference [VI, Kingma and Welling, 2014, Gal and Ghahramani, 2016, Blei et al., 2017] or complex sampling mechanisms through Markov chain Monte Carlo approaches [MCMC, Chen et al., 2014, Liu and Wang, 2016, Garriga-Alonso and Fortuin, 2021]. Deep ensembles [Dietterich, 2000, Lakshminarayanan et al., 2017] on the other hand maintain several independently initialized models to quantify predictive variance as uncertainty. Due to their simplicity and relative practical reliability, deep ensembles have become a widely established alternative to Bayesian approaches for uncertainty quantification in deep learning [Abdar et al., 2021].

However, both ensemble methods and approximate Bayesian methods typically incur substantial computational and memory costs, in particular for larger-scale models, motivating more efficient alternatives. RND [Burda et al., 2019] offers one such approach: by training a *predictor network* to mimic the outputs of a fixed, randomly initialized *target network*, RND produces a simple novelty or uncertainty signal via the squared prediction error. Random network distillation (RND) has seen empirical success in exploration, out-of-distribution detection, and continual learning [Burda et al., 2019, Nikulin et al., 2023, Matthews et al., 2024], yet the theoretical understanding of the nature of its uncertainty estimates remains blurry. In particular, it is unclear how—or whether—the RND error relates to the principled uncertainties produced for example by Bayesian inference or deep ensembles.

In this paper, we establish these missing theoretical connections by analyzing random network distillation in the idealized setting of infinite network width. In particular, we establish a Gaussian process (GP) interpretation of the self-predictive RND errors in the limit of infinitely wide neural networks, drawing on Neural Tangent Kernel (NTK) theory

[Jacot et al., 2018, Lee et al., 2020b].

Our three main contributions are:

1. *Ensemble equivalence with Standard RND*: We prove that, in the idealized infinite width limit, the squared prediction errors of standard RND coincide exactly with the variance of a deep ensemble.
2. *Posterior equivalence with Bayesian RND*: By engineering the RND target function, we design a *Bayesian RND* variant whose error distribution matches that of the exact Bayesian posterior predictive distribution of a neural network in the limit of infinite width.
3. *Posterior sampling with Bayesian RND*: Based on a multi-headed Bayesian RND model, we devise a posterior sampling algorithm that produces i.i.d. samples of the exact Bayesian posterior predictive distribution of neural networks in the limit of infinite width.

This unifying perspective on the uncertainty estimates produced by RND, deep ensembles, and Bayesian inference provides a novel understanding and theoretical support for the empirical effectiveness of RND and suggests avenues for future research directions towards principled Bayesian inference with minimal computational overhead.

2 PRELIMINARIES

We begin by establishing notation, defining RND formally, and briefly introducing the theoretical framework used in our analysis. In our analysis, we consider fully connected neural networks $f(x; \theta_t)$ of L layers of widths $n_1, \dots, n_L = n$, parametrized by θ_t at time t . The forward computation of such networks is defined recursively with $z_i^l(x; \theta_t^{\leq l})$ denoting the i -th output of layer l and

$$\begin{aligned} z_i^l(x, \theta_t^{\leq l}) &= \sigma_b b_i^l + \frac{\sigma_w}{\sqrt{n_{l-1}}} \sum_{j=1}^{n_{l-1}} w_{ij}^l x_j^l(x) \\ x_j^l(x) &= \phi(z_j^{l-1}(x; \theta_t^{\leq l-1})), \end{aligned} \quad (1)$$

where $\theta_t^{\leq l}$ denotes the parameters $\{w^1, b^1, \dots, w^l, b^l\}$ up to layer l , σ_b and σ_w denote scaling parameters of the forward computation, and $\phi: \mathbb{R} \rightarrow \mathbb{R}$ is a Lipschitz-continuous non-linearity. In Eq. 1, $n_0 = d_{in}$ and $x^1(x) = x$. The output of a scalar-output neural network is then given by $f(x; \theta_t) = z^L(x; \theta_t^{\leq L})$. We furthermore assume that parameters are initialized i.i.d. from a normal distribution $\theta_0 \sim \mathcal{N}(0, I)^1$. For convenience, we will sometimes overload notation to concatenate function outputs, for example indicating a set $\mathcal{X} = \{x_i \in \mathbb{R}^{d_{in}}\}_{i=1}^{N_D}$ and the corresponding function output as a column vector $f(\mathcal{X}; \theta_t) = (f(x_i; \theta_t))_{i=1}^{N_D}$, where

$f(\mathcal{X}; \theta_t) \in \mathbb{R}^{N_D \times K}$ or matrix-valued identities $\Sigma(\mathcal{X}, \mathcal{X}) = (\Sigma(x_i, x_j))_{i,j=1}^{N_D}$, where $\Sigma(\mathcal{X}, \mathcal{X}) \in \mathbb{R}^{N_D \times N_D}$. For conciseness our notation will furthermore use a shorthand for covariance and kernel matrices denoting $\Sigma_{\mathcal{X}\mathcal{X}} \equiv \Sigma(\mathcal{X}, \mathcal{X})$. In the following we briefly review methods pertinent to this work.

Random network distillation. Random network distillation [Burda et al., 2019] is an uncertainty quantification technique that employs two neural networks of identical architecture: A fixed, randomly initialized target network $g(x; \psi_0): \mathbb{R}^{d_{in}} \rightarrow \mathbb{R}^K$, and a *predictor network* $u(x; \vartheta_t)$, where parameters ϑ_t are subject to optimization via gradient descent. The predictor is trained to minimize the expected squared difference to the target network’s output on a set of data points $\mathcal{X} = \{x_i \in \mathbb{R}^{d_{in}}\}_{i=1}^{N_D}$

$$\mathcal{L}_{\text{rnd}}(\vartheta_t) = \frac{1}{2} \|u(\mathcal{X}; \vartheta_t) - g(\mathcal{X}; \psi_0)\|_2^2. \quad (2)$$

It is common to design RND with a multi-headed architecture with output dimension K and individual output heads $\{u_i(x; \vartheta_t)\}_{i=1}^K$, and $\{g_i(x; \psi_0)\}_{i=1}^K$, where the sum of squared prediction errors $\epsilon_i(x; \vartheta_t, \psi_0) = u_i(x; \vartheta_t) - g_i(x; \psi_0)$ at a test point x serves as an uncertainty signal

$$\epsilon^2(x; \vartheta_t, \psi_0) = \frac{1}{K} \sum_{i=1}^K (u_i(x; \vartheta_t) - g_i(x; \psi_0))^2. \quad (3)$$

Gaussian processes. In our analysis, we will frequently use GPs to model distributions over random functions: A univariate GP [Rasmussen and Williams, 2006] defines a distribution over functions $f^0 \sim \mathcal{GP}(\mu^0, \Sigma^0)$ characterized by a mean function $\mu^0: \mathbb{R}^{d_{in}} \rightarrow \mathbb{R}$ and a covariance (kernel) function $\Sigma^0: \mathbb{R}^{d_{in}} \times \mathbb{R}^{d_{in}} \rightarrow \mathbb{R}$ such that $f_0(\mathcal{X}_T)$ follows a multivariate Gaussian distribution $f_0(\mathcal{X}_T) \sim \mathcal{N}(\mu^0(\mathcal{X}_T), \Sigma^0(\mathcal{X}_T, \mathcal{X}_T))$ for any finite set of evaluation points $\mathcal{X}_T = \{x_i^{\text{Test}}\}_{i=1}^{N_T}$. We can condition a prior GP $\mathcal{N}(\mu^0(\mathcal{X}_T), \Sigma^0(\mathcal{X}_T, \mathcal{X}_T))$ on training data $\mathcal{X} = \{x_i\}_{i=1}^{N_D}$ and labels $\mathcal{Y} = \{y_i\}_{i=1}^{N_D}$ to obtain a posterior GP whose *posterior predictive distribution* is Gaussian with mean and covariance given by

$$\begin{aligned} \mu(\mathcal{X}_T) &= \mu^0(\mathcal{X}_T) + \Sigma_{\mathcal{X}_T\mathcal{X}}^0 (\Sigma_{\mathcal{X}\mathcal{X}}^0)^{-1} (\mathcal{Y} - \mu^0(\mathcal{X})), \\ \Sigma_{\mathcal{X}_T\mathcal{X}_T} &= \Sigma_{\mathcal{X}_T\mathcal{X}_T}^0 - \Sigma_{\mathcal{X}_T\mathcal{X}}^0 (\Sigma_{\mathcal{X}\mathcal{X}}^0)^{-1} \Sigma_{\mathcal{X}\mathcal{X}_T}^0. \end{aligned} \quad (4)$$

Learning dynamics with infinite width. We turn to analytical tools to establish solutions to the learning dynamics of neural networks in the limit of infinite width $n \rightarrow \infty$. Within this setting, we consider the training dynamics under *gradient flow*, the continuous-time limit of gradient descent $\frac{d}{dt} \theta_t = -\nabla_{\theta} \mathcal{L}(\theta_t)$. Under gradient flow with a square loss $\mathcal{L}(\theta_t) = \frac{1}{2} \|f(\mathcal{X}; \theta_t) - \mathcal{Y}\|_2^2$, the evolution of the NN f is described by a differential equation in function space

$$\begin{aligned} \frac{d}{dt} f(x; \theta_t) &= \nabla_{\theta} f(x; \theta_t)^{\top} \frac{d}{dt} \theta_t \\ &= -\nabla_{\theta} f(x; \theta_t)^{\top} \nabla_{\theta} f(\mathcal{X}; \theta_t) (f(\mathcal{X}; \theta_t) - \mathcal{Y}) \\ &\equiv -\Theta_t(x, \mathcal{X}) (f(\mathcal{X}; \theta_t) - \mathcal{Y}). \end{aligned} \quad (5)$$

¹Also known as NTK-parametrization, where variance scalings σ_b and σ_w affect both forward and gradient computations, yielding well-behaved gradients in the infinite-width limit.

The above learning dynamics are governed by a gradient similarity function, called the *neural tangent kernel* [NTK, Jacot et al., 2018], $\Theta_t(x, x') = \nabla_{\theta} f(x; \theta_t)^\top \nabla_{\theta} f(x'; \theta_t)$. While this inner product is dynamic and therefore intractable in general, the limit of infinite network width yields a remarkable simplification: (1) due to large number effects, the inner product kernel $\Theta_0(x, x')$ at initialization is deterministic despite the random initialization of θ_0 ; (2) $\Theta_t(x, x')$ remains constant throughout t under gradient flow [Jacot et al., 2018, Lee et al., 2020b]. In particular, this means $\lim_{n \rightarrow \infty} \Theta_0(x, x') = \lim_{n \rightarrow \infty} \Theta_t(x, x') \equiv \Theta(x, x')$ and converts Eq. 5 into a linear ordinary differential equation, which can be solved analytically. It can be shown that, under mild conditions, $f(x; \theta_t)$ converges to the kernel regression solution [see Jacot et al., 2018, and Appendix B.1]

$$f(x; \theta_\infty) = f(x; \theta_0) - \Theta_{x\mathcal{X}} \Theta_{\mathcal{X}\mathcal{X}}^{-1} (\mathcal{Y} - f(\mathcal{X}; \theta_0)), \quad (6)$$

Moreover, Lee et al. [2018] show that both $f(x; \theta_0)$ and $f(x; \theta_\infty)$ are indeed GPs described by the neural network Gaussian process [NNGP, Lee et al., 2018] $f(x; \theta_0) \sim \mathcal{GP}(0, \kappa_{xx'})$ and the converged GP defined in Theorem 2.1.

Theorem 2.1. [Lee et al., 2020b] (Distribution of post-convergence neural network functions) Let $f(\mathcal{X}_T; \theta_\infty)$ be a NN as defined in Eq.(1), and let $\mathcal{X}_T$ be testpoints. For random initializations $\theta_0 \sim \mathcal{N}(0, I)$, and in the limit $n \rightarrow \infty$, $f(\mathcal{X}_T; \theta_\infty)$ distributes as a Gaussian with mean and covariance given by

$$\begin{aligned} \mathbb{E}[f(\mathcal{X}_T, \theta_\infty)] &= \Theta_{\mathcal{X}_T \mathcal{X}} \Theta_{\mathcal{X} \mathcal{X}}^{-1} \mathcal{Y}, \\ \Sigma_{\mathcal{X}_T \mathcal{X}_T}^f(\theta_\infty) &= \kappa_{\mathcal{X}_T \mathcal{X}_T} + \Theta_{\mathcal{X}_T \mathcal{X}} \Theta_{\mathcal{X} \mathcal{X}}^{-1} \kappa_{\mathcal{X} \mathcal{X}} \Theta_{\mathcal{X} \mathcal{X}}^{-1} \Theta_{\mathcal{X} \mathcal{X}_T} \\ &\quad - (\Theta_{\mathcal{X}_T \mathcal{X}} \Theta_{\mathcal{X} \mathcal{X}}^{-1} \kappa_{\mathcal{X} \mathcal{X}_T} + h.c.), \end{aligned}$$

where *h.c.* is the Hermitian conjugate of the preceding term.

See also Appendix B.1.2 or Lee et al. [2020b]. Note that the GP described in Theorem 2.1 represents the law by which an infinite ensemble of infinitely wide neural networks from i.i.d. initializations distributes after training on $(\mathcal{X}, \mathcal{Y})$, but—as is—permits no Bayesian posterior interpretation, which is of the canonical form described in Eq. 4.

3 EQUIVALENCE OF RANDOM NETWORK DISTILLATION & DEEP ENSEMBLES

We proceed to characterize formally the relationship between the error signals as measured by random network distillation and the predictive variance of deep neural network ensembles. Before treating multivariate output dimensions in Section 3.1, we first consider scalar function outputs for simplicity, i.e. $f, u, g : \mathbb{R}^{d_{\text{in}}} \rightarrow \mathbb{R}$ with $K = 1$. This setup involves training a predictor $u(x; \vartheta_t)$ to match a fixed

random target function $g(x; \psi_0)$. Intuitively, the expected errors ought to vanish for training points in $\mathcal{X}$ and remain non-zero elsewhere, inheriting the randomness and generalization behaviors of the functions u and g . Owing to the linear training dynamics in the NTK regime, the dynamics of the error evolution $\frac{d}{dt} \epsilon(x; \vartheta_t, \psi_0)$ become akin to those outlined in Eq. 5 as

$$\begin{aligned} \frac{d}{dt} \epsilon(x; \vartheta_t, \psi_0) &= \nabla_{\theta} u(x; \vartheta_t)^\top \frac{d}{dt} \vartheta_t \\ &= -\nabla_{\vartheta} u(x; \vartheta_t)^\top \nabla_{\vartheta} \mathcal{L}_{\text{md}}(\vartheta_t) \\ &= -\Theta_t(x, \mathcal{X}) \epsilon(x; \vartheta_t, \psi_0). \end{aligned} \quad (7)$$

We then draw on the results of Theorem 2.1 to provide a probabilistic description of the self-predictive errors $\epsilon(x; \vartheta_\infty, \psi_0)$ of a converged RND model in the limit of infinite network width.

Theorem 3.1. (Distribution of post-convergence RND errors) Under NTK parametrization, let $u(x; \vartheta_\infty)$ be a converged prediction network in $t \rightarrow \infty$, with data $\mathcal{X}$ and fixed target network $g(\mathcal{X}; \psi_0)$. Let parameters ϑ_0, ψ_0 be drawn i.i.d. $\vartheta_0, \psi_0 \sim \mathcal{N}(0, I)$, with the resulting NNGP $u(x; \vartheta_0) \sim \mathcal{GP}(0, \kappa^u(x, x'))$ and $g(x; \psi_0) \sim \mathcal{GP}(0, \kappa^g(x, x'))$. The post-convergence RND error $\epsilon(\mathcal{X}_T; \vartheta_\infty, \psi_0)$ is Gaussian with zero mean and covariance

$$\begin{aligned} \mathbb{E}[\epsilon(\mathcal{X}_T, \vartheta_\infty, \psi_0)] &= 0, \\ \Sigma_{\mathcal{X}_T \mathcal{X}_T}^\epsilon(\vartheta_\infty, \psi_0) &= \kappa_{\mathcal{X}_T \mathcal{X}_T}^\epsilon + \Theta_{\mathcal{X}_T \mathcal{X}} \Theta_{\mathcal{X} \mathcal{X}}^{-1} \kappa_{\mathcal{X} \mathcal{X}}^\epsilon \Theta_{\mathcal{X} \mathcal{X}}^{-1} \Theta_{\mathcal{X} \mathcal{X}_T} \\ &\quad - (\Theta_{\mathcal{X}_T \mathcal{X}} \Theta_{\mathcal{X} \mathcal{X}}^{-1} \kappa_{\mathcal{X} \mathcal{X}_T}^\epsilon + h.c.), \end{aligned}$$

where $\kappa_{xx'}^\epsilon = \kappa_{xx'}^u + \kappa_{xx'}^g$, is the covariance kernel of initialization errors $\epsilon(x; \vartheta_0, \psi_0) = u(x; \vartheta_0) - g(x; \psi_0)$.

Proof sketch. The error function $u(x; \vartheta_\infty) - g(x; \psi_0)$ is a sum of the random post-convergence function $u(x; \vartheta_\infty)$ and the fixed random target function $g(x; \psi_0)$. The latter $g(x; \psi_0)$ is known to follow the NNGP. By the linearity of NTK learning dynamics, the online function $u(x; \vartheta_\infty)$ is an affine transformation of its initialization $u(x; \vartheta_0)$, which itself follows the NNGP. Moreover, this affine transformation is independent of g or ψ_0 , such that the error $\epsilon(x; \vartheta_\infty, \psi_0)$ is a sum of two independent GPs and therefore a GP itself. The resulting GP has zero-mean and covariance with an altered prior NNGP kernel $\kappa^\epsilon(x, x')$ composed of the online prior kernel $\kappa_{xx'}^u$ and the target prior kernel $\kappa_{xx'}^g$. See also Appendix B.1.3.

Corollary 3.2. (Equivalence in expectation between RND errors and ensemble variance) Under the conditions of Theorem 3.1, let $\epsilon(x; \vartheta_\infty, \psi_0)$ be the error function of a converged RND network with data $\mathcal{X}$. Moreover, for a regression problem on $\mathcal{X}$ for some labels $\mathcal{Y}$, let $\mathbb{V}[f(x; \theta_\infty)]$ denote the variance of converged NN functions random initializations. Furthermore, suppose an architectural equivalence between f , u , and g and i.i.d. parameter initialization

$\theta_0, \vartheta_0, \psi_0 \sim \mathcal{N}(0, I)$. The expected norm of the RND error $\epsilon^2(x; \vartheta_\infty, \psi_0)$ then coincides with the ensemble variance

$$\mathbb{E}_{\vartheta_0, \psi_0} [\epsilon^2(x; \vartheta_\infty, \psi_0)] = \mathbb{V}_{\theta_0} [f(x; \theta_\infty)]. \quad (8)$$

Proof sketch. Corollary 3.2 follows straightforwardly from Theorem 3.1 by using $\kappa^u(x, x') = \kappa^g(x, x')$. Taking the trace of the covariance matrix and dividing by 2, we recover the predictive ensemble variance $\mathbb{V}_{\theta_0} [f(x; \theta_\infty)]$.

Theorem 3.1 and Corollary 3.2 formally show that, for an architectural equivalence between ensemble, predictor and target network, the expected RND errors directly quantify the predictive variance of the corresponding infinite ensemble model described by Theorem 2.1. To the best of our knowledge, it is the first formal analysis of random network distillation in the NTK regime and reveals a first theoretical motivation for the popular algorithm: in the idealized infinite-width setting, *expected RND errors exactly quantify the variance of deep ensembles for any input x* .

3.1 MULTI-HEADED RANDOM NETWORK DISTILLATION

The analysis thus far has considered the *average* behavior of scalar network outputs for simplicity. While insightful in its own right, this setting does not reflect most common practical implementations of random network distillation and instead, if taken literally, would imply an ensemble of random network distillation models. To connect with common practical implementations that typically use multi-headed architectures for enhanced reliability and efficiency, we now seek to incorporate the probabilistic relation between different function outputs $f_i(x; \theta_t)$ and $f_j(x'; \theta_t)$ of a NN with shared hidden layers in the infinite-width limit. The result below identifies this relationship simply as a statistical independence between the different *random* network outputs $f_i(x; \theta_t)$ and $f_j(x'; \theta_t)$ for any time t during gradient flow optimization.

Proposition 3.3. (*Independence of NN functions*) Under NTK parametrization and in the limit $n \rightarrow \infty$, the random functions $f_i(x; \theta_t)$ of a NN with K output dimensions and shared hidden layers are mutually independent with covariance

$$\Sigma_{xx'}^{ij}(\theta_t) = \mathbb{E}[f_i(x; \theta_t) f_j(x'; \theta_t)] = \begin{cases} \Sigma_{xx'}^f(\theta_t) & i = j, \\ 0 & i \neq j, \end{cases}$$

on the interval $t \in [0, \infty)$.

Proof sketch. The property follows from known results that state the independence between output dimensions of the NNGP kernel κ and the NTK Θ [Arora et al., 2019, Lee et al., 2018, Jacot et al., 2018]. For both kernels, the proof proceeds by induction, where the independence property

between output dimensions is propagated layer-wise. The induction start is equal for both kernels, where first layer outputs, as well as gradients are linear transformations of the Gaussian first-layer weights. Both the NNGP and NTK permit a recursive formulation, through which the independence property can be propagated layer-wise, constituting the induction step. Combined with the learning dynamics of wide NNs, we can conclude that the individual function outputs of a multi-headed NN, too, are statistically independent for any time t on the interval $[0, \infty)$. See Appendix B.1.4 or Lee et al. [2018] and Jacot et al. [2018].

Notably, this decoupling holds despite the shared hidden layers and is an artifact of the learning dynamics exhibited in the infinite width limit and the NTK regime. In the absence of feature learning, output functions become statistically independent despite sharing a network body. By virtue of this independence property, a translation of the earlier obtained single-function results on RND error distributions (Theorem 3.1 and Corollary 3.2) to the multi-headed setting is straightforward. Our next result thus establishes an equivalence between the errors of the multi-headed RND algorithm, a widely used architecture in practice, and the variance of a finite-sized deep ensemble.

Theorem 3.4. (*Distributional equivalence between multi-headed RND and finite deep ensembles*) Under the conditions of Theorem 3.1, let $u_i(x; \vartheta_\infty), g_i(x; \psi_0)$ be the i -th output of predictor and target networks respectively with K output dimensions. Denote their sample mean RND error $\bar{\epsilon}^2(x; \vartheta_\infty, \psi_0) = \frac{1}{K} \sum_{i=1}^K \epsilon_i^2(x; \vartheta_\infty, \psi_0)$. Moreover, let $\{f(x; \theta_\infty^i)\}_{i=1}^{K+1}$ be an ensemble of $K+1$ NNs from i.i.d. initial draws θ_0 . Denote its sample variance $\bar{\sigma}_f^2(x; \theta_\infty^{1 \dots K+1}) = \frac{1}{K} \sum_{i=1}^{K+1} (f(x; \theta_\infty^i) - \frac{1}{K+1} \sum_{j=1}^{K+1} f(x; \theta_\infty^j))^2$. The sample mean RND error and sample ensemble variance distribute to the same law

$$\frac{1}{2} \bar{\epsilon}^2(x; \vartheta_\infty, \psi_0) \stackrel{D}{=} \bar{\sigma}_f^2(x; \theta_\infty^{1 \dots K+1}), \quad (9)$$

where $\stackrel{D}{=}$ indicates an equality in distribution, namely by a scaled Chi-squared distribution $\bar{\sigma}_f^2(x; \theta_\infty^{1 \dots K+1}) \sim \frac{\Sigma_{xx}^f(\theta_\infty)}{K} \chi^2(K)$ with scale $\Sigma_{xx}^f(\theta_\infty)$ given by the analytical variance as given in Theorem 2.1.

Proof sketch. By Proposition 3.3, the function heads $\{u_i(x; \vartheta_\infty)\}_{i=1}^K$ are K independent predictors, each trained to match their independent targets $g_i(x; \psi_0)$. Thus, the errors $\{\epsilon_i(x; \vartheta_\infty, \psi_0)\}_{i=1}^K$ are i.i.d. samples from the error distribution outlined in Proposition 3.2. In particular, $\bar{\epsilon}^2$ is the empirical mean of i.i.d. samples from a Gaussian which is known to be Chi-squared distributed. Similarly, we have that the ensemble $\{f(x; \theta_\infty^i)\}_{i=1}^{K+1}$ are $K+1$ i.i.d. samples from the GP defined in Theorem 2.1, again yielding the known Chi-squared distribution for its sample variance $\bar{\sigma}_f^2(x; \theta_\infty^{1 \dots K+1})$. See Appendix B.1.5.

Theorem 3.4 establishes a distributional equality between the empirical error of a multi-headed RND architecture and the empirical variance of a finite ensemble of neural networks in the limit of infinite width, providing a theoretical motivation for the use of RND and its common multi-headed architecture as an uncertainty quantification technique.

In a broader sense, we believe this analysis is insightful to many practitioners using random network distillation by establishing an intuitive link between theory and practice. Still, the NTK-based perspective applies to an inherently idealized regime and naturally opens up new avenues for investigation. Understanding the relationship between RND networks and deep ensembles at finite width, where feature learning impacts behavior, remains a critical open question beyond the scope of our current framework. Yet, intriguing possibilities also arise within the infinite-width setting itself: Could the properties of the RND target network be deliberately chosen or modified? Exploring different target initializations offers a computationally inexpensive lever to shape the uncertainty signal captured by RND. Indeed, pursuing this very direction, the next section investigates how a specific adaptation of the RND target function allows us to establish a direct correspondence not just with ensemble variance, but with the principled uncertainty quantification provided by Bayesian posterior inference.

4 EQUIVALENCE OF RANDOM NETWORK DISTILLATION & BAYESIAN POSTERIORES

Having formulated an equivalence between standard random network distillation and deep ensemble variance, we now proceed to investigate how theoretical connections to the Bayesian inference framework can be established by invoking deliberate changes to the standard random network distillation algorithm, namely by modifying the fixed target function g . Our goal is to show that the RND error signal itself can, under specific conditions, be interpreted as a draw from a centered Bayesian posterior predictive distribution.

To this end, we briefly recall Bayesian inference with the classical Gaussian linear model. We define a regression model as $f(x; \theta) = \phi(x)^\top \theta$ with a feature mapping $\phi : \mathbb{R}^{d_{\text{in}}} \rightarrow \mathbb{R}^{d_P}$, and a prior distribution over the parameters $p(\theta) \sim \mathcal{N}(0, \Sigma^0)$. The prior distribution $p(\theta)$ implicitly defines a GP prior $f^0(x; \theta) \sim \mathcal{GP}(0, \phi(x)^\top \Sigma^0 \phi(x'))$, with the prior kernel $K_{xx'} = \phi(x)^\top \Sigma^0 \phi(x')$. Within this linear model², we look to infer a posterior distribution over functions given observations $\mathcal{X} = \{x_i \in \mathbb{R}^{d_{\text{in}}}\}_{i=1}^{N_D}$ and labels $\mathcal{Y} = \{y_i \in \mathbb{R}\}_{i=1}^{N_D}$. Owing to our prior choice, the corre-

sponding *posterior predictive* distribution conditioned on $\mathcal{X}, \mathcal{Y}$ is a GP with

$$p(f|x, \mathcal{X}, \mathcal{Y}) \sim \mathcal{N}(K_{x\mathcal{X}} K_{\mathcal{X}\mathcal{X}}^{-1} \mathcal{Y}, K_{xx} - K_{x\mathcal{X}} K_{\mathcal{X}\mathcal{X}}^{-1} K_{\mathcal{X}x}). \quad (10)$$

When contrasting this identity with the GP governing the distribution of converged NN functions of Theorem 3.1, one observes a disparity in the structure of the covariance functions. While Theorem 2.1 and Theorem 3.1, too, specify GPs, they do not permit an interpretation as a Bayesian posterior predictive distribution [Lee et al., 2020b] due to the presence of two (in general) distinct kernel functions, namely the NNGP kernel κ and the NTK Θ . However, inspection of Theorem (3.1) and Eq. 10 suggests a path: if the prior kernel components within $\Sigma_{xx'}^\epsilon(\vartheta_\infty, \psi_0)$, namely $\kappa_{xx'}^\epsilon$, are aligned with the dynamics kernel $\Theta_{xx'}$ (i.e., if $\kappa^\epsilon \propto \Theta$), then the resulting covariance structure simplifies to the desired Bayesian posterior form of

$$f(x; \theta_\infty) \sim \mathcal{N}(\Theta_{x\mathcal{X}} \Theta_{\mathcal{X}\mathcal{X}}^{-1} \mathcal{Y}, \Theta_{xx} - \Theta_{x\mathcal{X}} \Theta_{\mathcal{X}\mathcal{X}}^{-1} \Theta_{\mathcal{X}x}). \quad (11)$$

An important insight here is that Eq. 11 now is the *exact Bayesian posterior predictive distribution of a neural network in the infinite width limit*, which corresponds to a kernel regression model with the NTK as a GP prior $\mathcal{GP}(0, \Theta_{xx'})$ and conditioned on the data $(\mathcal{X}, \mathcal{Y})$.

The idea of aligning the prior and dynamic kernels has been previously explored by He et al. [2020] to construct *Bayesian ensembles* where the predictive distribution of the ensemble matches the posterior predictive distribution of the *NTK-GP*. We propose that a similar alignment can be achieved in the RND framework by constructing the target function $g(x; \psi_0)$ to assume a specific form. The idea is to design a target $\tilde{g}(x; \vartheta_0, \psi_0)$ such that when a predictor $u(x; \vartheta_0)$ is trained to match it, the resulting “Bayesian” error distribution $\epsilon^b(x; \vartheta_\infty, \vartheta_0, \psi_0) = u(x; \vartheta_\infty) - \tilde{g}(x; \vartheta_0, \psi_0)$ behaves like a draw from the posterior of a Bayesian model whose prior kernel is the NTK $\Theta_{xx'}$ itself³.

In the random network distillation algorithm, the prior kernel $\kappa_{xx'}^{\epsilon^b}$ of initialization errors $\epsilon^b(x; \vartheta_0, \vartheta_0, \psi_0) = u(x; \vartheta_0) - \tilde{g}(x; \vartheta_0, \psi_0)$ is given by the sum of the online prior kernel and the target prior kernel $\kappa_{xx'}^{\epsilon^b} = \kappa_{xx'}^u + \kappa_{xx'}^{\tilde{g}}$ (cf. Theorem 3.1), provided that u and $\tilde{g}$ follow independent GPs. To obtain an error prior kernel that aligns with the NTK such that $\kappa_{xx'}^{\epsilon^b} = \Theta_{xx'}$, one may thus construct the target prior such that it satisfies $\kappa_{xx'}^{\tilde{g}} = \Theta_{xx'} - \kappa_{xx'}^u$. To this end, a closer inspection of the relation between the NNGP kernel $\kappa_{xx'}^u$ and the NTK $\Theta_{xx'}$ is instructive. For this purpose, we will view the online network $u(x; \vartheta_0)$ as a random feature model with its forward computation path

²We use a noise-free regression model for ease of notation here, but extensions to the noisy case by including an observation noise term $\sigma_n^2 I$ in the kernel matrix inversions (cf. Eq. 10, 11) are straightforward.

³The newly constructed target function $\tilde{g}(x; \vartheta_0, \psi_0)$ uses both ϑ_0 and ψ_0 for reasons that will become clear in the remainder of section.

as described in Eq. 1. Let in this scenario $x^L(x)$ denote the output vector, or the post-activations, before the final linear layer and denote the last-layer parameters at initialization $t = 0$ as (w^L, b^L) . We can write the NN output at initialization $u(x; \vartheta_0)$ as

$$u(x; \vartheta_0) = \sigma_b b^L + \frac{\sigma_w}{\sqrt{n_{L-1}}} \sum_{i=1}^{n_{L-1}} w_i^L x_i^L(x), \quad (12)$$

that is, as a simple linear model of the random final post-activations $x^L(x)$. Viewing the function in Eq. 12 as a random feature model leads to a central insight: since the last-layer weights and biases (w^L, b^L) are assumed to be initialized i.i.d. from a standard normal $(w^L, b^L) \sim \mathcal{N}(0, I)$, Eq. 12 describes a (random) affine transformation of a Gaussian vector⁴ whose covariance in the limit $n \rightarrow \infty$ is quantified by the NNGP kernel $\kappa_{xx'}^u$ given by

$$\kappa_{xx'}^u = \mathbb{E}[u(x; \vartheta_0)u(x'; \vartheta_0)] = \sigma_b^2 + \sigma_w^2 \mathbb{E}[x_i^L(x)x_i^L(x')]. \quad (13)$$

Let us now compare this expression for the prior kernel $\kappa_{xx'}^u$ of the online network with its dynamics kernel $\Theta_{xx'}$. In particular, we will split the dynamics kernel $\Theta_{xx'}$ into a last-layer component $\Theta_{xx'}^L = \nabla_{\{w^L, b^L\}} u(x; \vartheta_0)^\top \nabla_{\{w^L, b^L\}} u(x'; \vartheta_0)$ and a component summarizing all preceding parameters $\Theta_{xx'}^{\leq L-1} = \nabla_{\vartheta_{\leq L-1}} u(x; \vartheta_0)^\top \nabla_{\vartheta_{\leq L-1}} u(x'; \vartheta_0)$ such that $\Theta_{xx'} = \Theta_{xx'}^L + \Theta_{xx'}^{\leq L-1}$. Since $u(x; \vartheta_0)$ is linear in the last-layer parameters $\{w^L, b^L\}$ (cf. Eq. 12), we make the crucial observation that the last-layer NTK component $\Theta_{xx'}^L$ equals the NNGP prior kernel $\Theta_{xx'}^L = \kappa_{xx'}^u$ ⁵. This property gives a clear instruction for engineering the prior kernel of the target network: by constructing $\kappa_{xx'}^{\tilde{g}}$ such that $\kappa_{xx'}^{\tilde{g}} = \Theta_{xx'}^{\leq L-1}$ and independently from $\kappa_{xx'}^u$, we obtain an error prior as

$$\kappa_{xx'}^{\epsilon} = \kappa_{xx'}^{\tilde{g}} + \kappa_{xx'}^u = \Theta_{xx'}^L + \Theta_{xx'}^{\leq L-1} = \Theta_{xx'}. \quad (14)$$

In the following, we will thus aim to construct a target function $\tilde{g}(x; \vartheta_0, \psi_0)$ with the desired property $\kappa_{xx'}^{\tilde{g}} = \Theta_{xx'}^{\leq L-1}$,

⁴To see the correspondence in Eq. 13, first notice that due to the i.i.d. initialization of (w^L, b^L) , any cross-products (e.g., involving elements indexed with $i \neq j$) vanish in the expectation $\mathbb{E}[u(x; \vartheta_0)u(x'; \vartheta_0)]$. The expectation thus becomes $\mathbb{E}[u(x; \vartheta_0)u(x'; \vartheta_0)] = \mathbb{E}_{w \leq L, b \leq L} [\sigma_b^2 + \frac{\sigma_w^2}{n_{L-1}} \sum_{i=1}^{n_{L-1}} x_i^L(x)x_i^L(x')]$. By linearity, the expectation on the r.h.s. can be pulled inside the sum and by symmetry we have that $\mathbb{E}_{w \leq L, b \leq L} [x_i^L(x)x_i^L(x')]$ is independent of i , s.t. $\mathbb{E}_{w \leq L, b \leq L} [\frac{\sigma_w^2}{n_{L-1}} \sum_{i=1}^{n_{L-1}} x_i^L(x)x_i^L(x')] = \sigma_w^2 \mathbb{E}[x_i^L(x)x_i^L(x')]$.

⁵To see this correspondence, notice that the last-layer gradient inner product $\nabla_{\{w^L, b^L\}} u(x; \vartheta_0)^\top \nabla_{\{w^L, b^L\}} u(x'; \vartheta_0)$ reduces to the sum $\sigma_b^2 + \frac{\sigma_w^2}{n_{L-1}} \sum_{i=1}^{n_{L-1}} x_i^L(x)x_i^L(x')$, where the r.h.s. sum tends to its expectation in the limit $n_{L-1} \rightarrow \infty$ given that summands are identically distributed (as before by symmetry) and independent (which is shown more rigorously for example in Sec. B.1.4).

in particular by modeling $\tilde{g}$ as a linear function in the feature space corresponding to gradients in earlier layers. This approach has also previously been explored by He et al. [2020] to obtain Bayesian ensembles.

Proposition 4.1. (*Bayesian RND target function*) Under the conditions of Theorem 3.1, let $u(x; \vartheta_0)$ and $g(x; \psi_0)$ be neural networks of L layers with parameters $\vartheta_0, \psi_0 \sim \mathcal{N}(0, I)$ i.i.d. Moreover, let $\psi_0^L = \{w^L, b^L\}$ denote the last-layer parameters of ψ_0 and $\psi_0^{\leq L-1}$ the parameters of all preceding layers. Suppose the target function $\tilde{g}(x; \vartheta_0, \psi_0)$ is given by

$$\tilde{g}(x; \vartheta_0, \psi_0) = \nabla_{\vartheta_0} u(x; \vartheta_0)^\top \psi_0^*,$$

where $\psi_0^* = \{\psi_0^{\leq L-1}, 0_{\dim(\psi_0^L)}\}$ is a copy of ψ_0 with its last-layer weights set to 0. In the infinite width limit $n \rightarrow \infty$, $\tilde{g}(x; \vartheta_0, \psi_0)$ distributes by construction as $\tilde{g}(x; \vartheta_0, \psi_0) \sim \mathcal{GP}(0, \kappa_{xx'}^{\tilde{g}})$ where $\kappa_{xx'}^{\tilde{g}} = \Theta_{xx'}^{\leq L-1}$.

Proof sketch. The function $\tilde{g}(x; \vartheta_0, \psi_0)$ is by construction equivalent to a linear function with the (random) feature map $\nabla_{\vartheta_{\leq L-1}} u(x; \vartheta_0)$ given by the gradient of parameters in the pre-final layers and with a parameter vector $\psi_0^{\leq L-1}$. Conditioned on ϑ_0 , the random function $\tilde{g}(x; \vartheta_0, \psi_0)$ is thus an affine transformation of the Gaussian vector $\psi_0^{\leq L-1}$ and thus a GP itself, at any width n . Using the central results by Jacot et al. [2018] that $\Theta_{0,xx'} \rightarrow \Theta_{xx'}$ as $n \rightarrow \infty$ and appealing to the bounded convergence theorem, the limiting distribution of the *unconditioned* random function $\tilde{g}(x; \vartheta_0, \psi_0)$, too, becomes Gaussian with the deterministic covariance $\Theta_{xx'}^{\leq L-1}$.

While the specific form of the kernel $\Theta_{xx'}^{\leq L-1} = \Theta_{xx'} - \Theta_{xx'}^L$ seems unusual as a standalone prior, it is crucially important in shaping the final error distribution. This is because with the altered ‘‘Bayesian’’ target function $\tilde{g}(x; \vartheta_0, \psi_0)$ we can shape the covariance structure of errors at initialization by satisfying Eq. 14, appealing to Theorem (3.1). With the engineered target function $\tilde{g}(x; \vartheta_0, \psi_0)$, the learning dynamics of an RND model where the predictor network $u(x; \vartheta_t)$ learns to mimic $\tilde{g}(\mathcal{X}; \vartheta_0, \psi_0)$ can be shaped in the desired way. Our central statement is that the distribution of the error between the converged predictor $u(x; \vartheta_\infty)$ and the target function $\tilde{g}(x; \vartheta_0, \psi_0)$ will then no longer reflect the variance of deep ensembles trained with gradient descent, but will instead directly exhibit the statistics of a Bayesian posterior predictive distribution derived from the NTK-GP prior. Theorem 4.2 formalizes this result.

Theorem 4.2. (*Distribution of Bayesian RND errors*) Under the conditions of Theorem 3.1, let $u(x; \vartheta_\infty)$ be a converged predictor network trained on data $\mathcal{X}$ with labels from the fixed target function $\tilde{g}(\mathcal{X}; \vartheta_0, \psi_0)$ as defined in Proposition 4.1. Let parameters ϑ_0, ψ_0 be drawn i.i.d. $\vartheta_0, \psi_0 \sim \mathcal{N}(0, I)$. The converged Bayesian RND error

$\epsilon^b(\mathcal{X}_T; \vartheta_\infty, \vartheta_0, \psi_0) = u(\mathcal{X}_T; \vartheta_\infty) - \tilde{g}(\mathcal{X}_T; \vartheta_0, \psi_0)$ on a test set $\mathcal{X}_T$ is Gaussian with zero mean and covariance

$$\Sigma_{\mathcal{X}_T \mathcal{X}_T}^{\epsilon^b}(\vartheta_\infty, \vartheta_0, \psi_0) = \Theta_{\mathcal{X}_T \mathcal{X}_T} - \Theta_{\mathcal{X}_T \mathcal{X}} \Theta_{\mathcal{X} \mathcal{X}}^{-1} \Theta_{\mathcal{X} \mathcal{X}_T},$$

and thus recovers the covariance of the exact Bayesian posterior predictive distribution of an infinitely wide neural network with the corresponding NTK $\Theta_{xx'}$.

Proof sketch. The result follows by combining Theorem 3.1 and Proposition 4.1, provided that the GP governing the predictor initialization $\kappa_{xx'}^u$ and the target function $\kappa_{xx'}^{\tilde{g}}$ are independent. Owing to the fact that the parameters ϑ_0 and ψ_0 are drawn independently, the independence between $u(x; \vartheta_0)$ and $\tilde{g}(x; \vartheta_0, \psi_0)$ is apparent by rewriting the covariance $\mathbb{E}[u(x; \vartheta_0) \tilde{g}(x; \vartheta_0, \psi_0)]$ in terms of conditional expectations on ϑ_0 by the law of total expectation. Furthermore, since $\Theta_{xx'} = \Theta_{xx'}^L + \Theta_{xx'}^{\leq L-1}$ and $\kappa_{xx'}^{\tilde{g}} = \Theta_{xx'}^{\leq L-1}$, $\kappa_{xx'}^u = \Theta_{xx'}^L$, we have that $\kappa_{xx'}^{\epsilon^b} = \Theta_{xx'}$. In other words, the GP kernel of initial errors aligns with the NTK of the on-line predictor, such that the distribution of post-convergence errors in Theorem 3.1 simplifies significantly. This same covariance function indeed also defines the posterior predictive distribution of infinitely wide neural networks as described by the GP with prior $\mathcal{GP}(0, \Theta_{xx'})$ and conditioned on $(\mathcal{X}, \mathcal{Y})$.

Theorem 4.2 shows that with a specifically engineered target function, the RND error signal $\epsilon^b(x; \vartheta_\infty, \vartheta_0, \psi_0) = u(x; \vartheta_\infty) - \tilde{g}(x; \vartheta_0, \psi_0)$ is no longer just related to ensemble variance, but rather becomes a direct sample from the centered posterior predictive distribution of a Bayesian model whose prior kernel is the NTK itself. This novel result provides a direct bridge between RND and Bayesian inference in the limit of infinite network width, providing a useful insight: the error signal generated by this modified RND procedure is not merely a heuristic measure of distance, but is itself a random draw from the (centered) Bayesian posterior predictive distribution of an NTK-based GP. This direct distributional equivalence has immediate practical implications, for example prescribing rather straightforwardly how this Bayesian form of RND can be used for exact posterior sampling. By applying Proposition 3.3 to the multi-headed Bayesian RND architecture⁶, in contrast to obtaining samples from deep ensembles as done in Theorem 3.4, we now obtain several independent samples from the centered posterior predictive distribution through $\epsilon_i^b(x; \vartheta_\infty, \vartheta_0, \psi_0) = u_i(x; \vartheta_\infty) - \tilde{g}_i(x; \vartheta_0, \psi_0)$. The below corollary details how this can be leveraged to conduct a posterior sampling procedure, requiring access only to a mean estimate and a single Bayesian RND model.

⁶In a multi-headed architecture, the Bayesian target function described in Proposition 4.2 becomes a Jacobian-vector product (JVP). Several common machine learning libraries (e.g., JAX [Bradbury et al., 2018]) offer dedicated algorithms to compute such JVPs efficiently.

Corollary 4.3 (Posterior Sampling via Bayesian RND). *Let $\mathcal{N}(\mu^b(x), \Sigma_{xx'}^b)$ be the posterior predictive distribution of an infinitely wide neural network conditioned on x with mean $\mu^b(x) = \Theta_{x\mathcal{X}} \Theta_{\mathcal{X}\mathcal{X}}^{-1} \mathcal{Y}$ and covariance $\Sigma_{xx'}^b = \Theta_{xx'} - \Theta_{x\mathcal{X}} \Theta_{\mathcal{X}\mathcal{X}}^{-1} \Theta_{\mathcal{X}x'}$. Suppose $\tilde{\mu}(x; \theta_\infty) \approx \mu^b(x)$ is an estimate of the mean function and let $\{\epsilon_i^b(x; \vartheta_\infty, \vartheta_0, \psi_0)\}_{i=1}^K$ be error functions of a K -head Bayesian RND model as defined in Theorem 4.2.*

The following procedure generates (at most K) independent samples from the conditional posterior predictive distribution $\mathcal{N}(\mu^b(x), \Sigma_{xx'}^b)$:

1. sample $i \sim \mathcal{U}[1, K]$
2. compute $\tilde{\mu}_i(x) = \tilde{\mu}(x; \theta_\infty) + \epsilon_i^b(x; \vartheta_\infty, \vartheta_0, \psi_0)$
3. $\tilde{\mu}_i(x)$ is an i.i.d. sample from the conditional posterior predictive $\mathcal{N}(\mu^b(x), \Sigma_{xx'}^b)$

Proof sketch. The result follows directly from Theorem 4.2 and application of the independence argument of Proposition 3.3 to the multi-headed setting.

Corollary 4.3 shows that, given an estimator of the posterior predictive mean, a modified Bayesian RND setup can be used to perform direct Bayesian posterior sampling in the NTK limit. By extension, this offers a pathway to performing exact Bayesian inference through the lens of network distillation, provided that the target and predictor networks initializations are handled deliberately.

This completes our theoretical development, first showing an equivalence of RND in the NTK regime to ensemble variance and now, through specific modifications to its target function, to the generation of independent samples from exact Bayesian posterior predictive distributions.

5 NUMERICAL ANALYSIS

We proceed with a numerical analysis to validate the thus far presented results.

Code for reproduction of all experiments is available at <https://github.com/anyboby/rnd-bayes-ensemble-equivalence>.

In the following, we study how predictive RND errors relate to predictive variances of deep ensembles in practice, both in the standard and Bayesian settings. To this end, we train two-layer connected neural networks with SiLU activations [Elfwing et al., 2018] on a synthetic dataset with $N = 10$ train and $\tilde{N} = 5000$ test samples from an isotropic Gaussian $x_i \sim \mathcal{N}(0, I_3)$. Ensemble models are fit to a toy target function, and multiheaded RND models optimized as described above. The variance of the true underlying GP is approximated with Monte-Carlo estimates of 512 independent models and a single Bayesian RND model with 512 heads, such that a small residual amount of discrepancy is to be

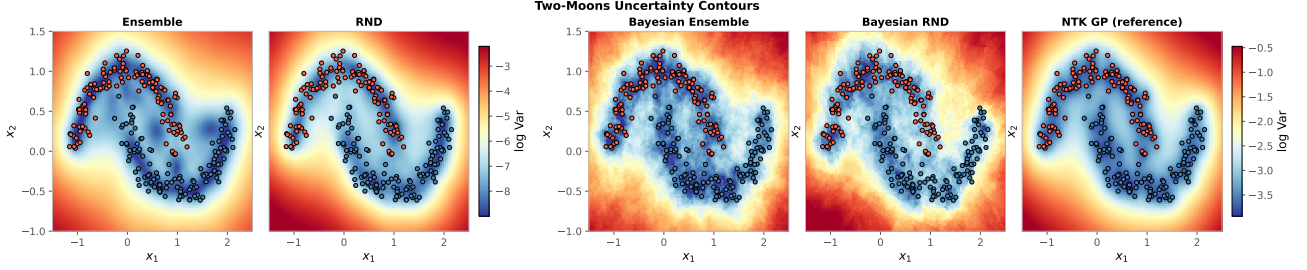


Figure 1: Predictive uncertainty contours on two-moons at width $n = 1024$ (logarithmic predictive variance). RND and standard ensembles, as well as Bayesian RND and Bayesian ensembles closely match each other respectively. Bayesian variants recover the NTK-GP posterior predictive variance more accurately in scale but exhibit more noise.

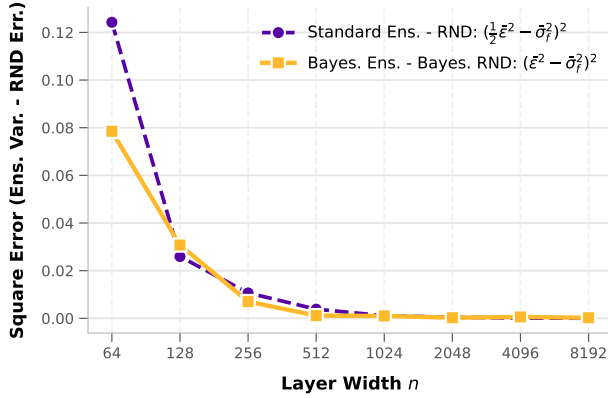


Figure 2: Test-set errors between predictive variances of (Bayesian) ensembles and self-predictive errors of (Bayesian) RND vanish with large layer widths.

expected. Fig. 2 shows a stark decrease in average squared discrepancy between test evaluations of predictive ensemble variances and RND errors as model width increases, a trend in line with our theoretical derivations and present even at practical network widths. Further evaluations and details of this experiment are reported in Appendix C.

Two-Moons. We further evaluate the proposed equivalences on the two-moons dataset [Pedregosa et al., 2011], a standard nonlinear binary classification dataset, which we cast as a regression problem with targets $y \in \{-1, +1\}$ to remain consistent with the regression setting underlying Theorem 4.2. Fig. 1 shows predictive uncertainty contours at a realistic width of $n = 1024$ and ensemble sizes $K = 64$ for RND, standard ensembles, Bayesian RND, Bayesian ensembles, and the NTK-GP posterior variance as ground truth. Even at limited width, both RND and Bayesian RND closely mirror their ensemble counterparts. The Bayesian variations more closely recover the overall higher NTK-GP variance in scale, but exhibit more noise than the standard models. We moreover confirm that the same width-dependent vanishing

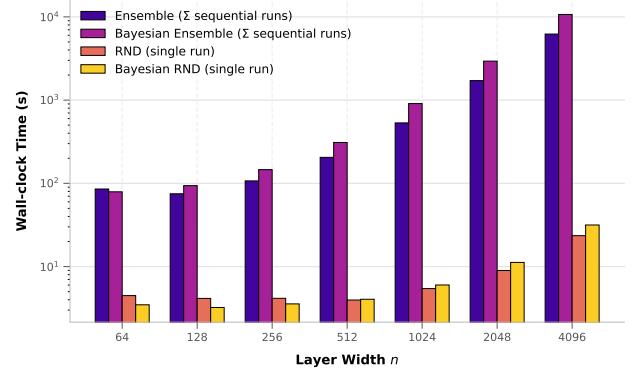


Figure 3: Wall-clock training time vs. network width on two-moons. RND retains a clear computational advantage over deep ensembles, which require sequential chunking to fit in memory at large widths. Bayesian variants add small overhead over their standard counterparts.

discrepancy between RND/ensembles and their Bayesian counterparts observed in Fig. 2 persists on this dataset, and across alternative initialization schemes (truncated Gaussian, uniform, orthogonal), as well as when replacing full-batch gradient descent with minibatched stochastic gradient descent; see Appendix C for these additional results.

Finally, we report wall-clock training times against network width for RND, Bayesian RND, standard ensembles, and Bayesian ensembles on the Two-Moons task in Fig. 3. As ensembles of size 512 exceeded our available memory at the largest widths considered, we run them sequentially in chunks of 16 and sum the resulting wall-clock time; this illustrates a memory–runtime trade-off that disadvantages full ensemble approaches at scale, whereas RND retains its computational edge throughout by requiring only a single network regardless of the targeted ensemble size. Notably, the Bayesian variants of both RND and ensembles add only marginal computational overhead over their standard counterparts, as the additional target evaluation (a single JVP)

is cheap relative to a full forward-backward pass, making Bayesian RND an attractive lightweight alternative to standard or Bayesian deep ensembles alike.

6 RELATED WORK

A substantial body of research studies the analytical learning dynamics of deep learning, particularly in the infinite-width limit. Central to our analysis are seminal works characterizing the NNGP [Lee et al., 2018] at initialization, the dynamics-governing NTK [Jacot et al., 2018], and the evolution of wide networks as linear models [Lee et al., 2020b, Arora et al., 2019, Chizat and Bach, 2018]. This provides a theoretical framework for analytical descriptions of deep ensembles [Lakshminarayanan et al., 2017, Dietterich, 2000], with subsequent studies using NTK theory to precisely characterize ensemble variances under various conditions, including observation noise [Yang, 2019, Kobayashi et al., 2022, Calvo-Ordoñez et al., 2024].

A central line of work for our paper is the connection between deep ensembles and Bayesian inference in infinite-width NTK regime. Notably, He et al. [2020] demonstrate how to construct “Bayesian ensembles”, an approach we adopt to construct “Bayesian RND” algorithms. The broader link between deep ensembles and approximations of Bayesian posteriors has been studied extensively [Khan et al., 2019, Osawa et al., 2019, D’Angelo and Fortuin, 2021, Osband et al., 2019, Izmailov et al., 2021].

More recently, NTK-based approaches have been used for single-model uncertainty estimation [Zanger et al., 2026a] or ad-hoc uncertainty quantification [Wilson et al., 2025]. Our work provides a theoretical basis for RND [Burda et al., 2019], which belongs to a class of computationally cheaper, single-model methods [Pathak et al., 2017, Lahlou et al., 2021, Guo et al., 2022, Sensoy et al., 2018, Van Amersfoort et al., 2020, Rudner et al., 2022, Laurent et al., 2022, Tagasovska and Lopez-Paz, 2019].

Moreover, uncertainty quantification from the lens of learning dynamics is moreover widespread in reinforcement learning (RL) [Xiao et al., 2021, Cai et al., 2019, Wai et al., 2020, Lyle et al., 2022, Yang et al., 2020], the original application domain of RND. Notably, Zanger et al. [2026b] derive an RND-like estimator for value function uncertainty using NTK theory. More broadly, deep ensembles and Bayesian methods are widely used in RL, driving exploration [Osband et al., 2016, Chen et al., 2017, Osband et al., 2019, Nikolov et al., 2019, Ishfaq et al., 2021, Zanger et al., 2024].

7 CONCLUSIONS

In this work, we have established a novel theoretical understanding of random network distillation (RND) by connecting it to the principled uncertainty frameworks of deep

ensembles and Bayesian inference. By analyzing these techniques within the unifying setting of infinitely wide neural networks, we provide a clear analytical interpretation for the empirically successful RND algorithm.

Our analysis yields a twofold equivalence: first, we prove that the squared error of standard RND exactly recovers the predictive variance of deep ensembles in the NTK regime. Second, we demonstrate that the RND framework is more versatile; by deliberately designing the RND target function, the resulting error signal can be made to directly mirror the centered posterior predictive distribution of an NTK-governed GP, that is, the exact posterior predictive distribution of neural networks in the infinite width limit. This “Bayesian RND” variant furthermore allows for posterior sampling procedures that produce i.i.d. samples from this posterior. Our work thereby unifies RND, ensembles, and Bayesian inference under the same theoretical lens from an infinite width perspective.

Crucially, our findings hold under the assumptions infinite-width and the NTK regime, a setting where networks effectively linearize and operate as kernel machines with a fixed kernel. This “lazy” training regime, while analytically tractable and predictive for very wide networks, does not capture the phenomenon of feature learning. The degree to which our established equivalences translate to practical, finite-width networks that learn features remains a significant open question.

Conversely, this also suggests avenues for future research: deviations between RND, ensembles, and Bayesian posteriors in practice must arise from departures from the NTK regime. Characterizing these deviations specifically could lead to novel techniques and a deeper understanding of computationally efficient approaches that approximate Bayesian inference, operating well outside the kernelized infinite-width setting. Another exciting direction is the concept of target engineering as a cheap way of studying priors for Bayesian deep learning, an actively studied field that garners widespread interest from the uncertainty quantification and Bayesian deep learning community.

Acknowledgements

This project has received funding from the EU Horizon 2020 programme *Epistemic AI* under grant number 964505.

References

- Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U. Rajendra Acharya, Vladimir Makarenkov, and Saeid Nahavandi. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *arXiv:2011.06225*, 2021.
- Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *Advances in neural information processing systems*, 32, 2019.
- David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical association*, 112(518):859–877, 2017.
- James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Nectala, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL <http://github.com/jax-ml/jax>.
- Yuri Burda, Harrison Edwards, Amos J. Storkey, and Oleg Klimov. Exploration by random network distillation. In *International conference on learning representations*, 2019.
- Qi Cai, Zhuoran Yang, Jason D Lee, and Zhaoran Wang. Neural temporal-difference learning converges to global optima. *Advances in neural information processing systems*, 32, 2019.
- Sergio Calvo-Ordoñez, Konstantina Palla, and Kamil Ciosek. Epistemic uncertainty and observation noise with the neural tangent kernel. *arXiv preprint arXiv:2409.03953*, 2024.
- Yuan Cao and Quanquan Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. *Advances in neural information processing systems*, 32, 2019.
- Richard Y Chen, Szymon Sidor, Pieter Abbeel, and John Schulman. UCB exploration via Q-ensembles. *arXiv preprint arXiv:1706.01502*, 2017.
- Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pages 1683–1691. PMLR, 2014.
- Lenaïc Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. *Advances in neural information processing systems*, 31, 2018.
- Francesco D’Angelo and Vincent Fortuin. Repulsive deep ensembles are bayesian. *Advances in Neural Information Processing Systems*, 34:3451–3465, 2021.
- Thomas G Dietterich. Ensemble methods in machine learning. In *Multiple classifier systems: First international workshop, MCS*. Springer, 2000.
- Rick Durrett. *Probability: Theory and examples*, volume 49. Cambridge university press, 2019.
- Stefan Elfving, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107: 3–11, 2018.
- Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International conference on machine learning*, pages 1050–1059. PMLR, 2016.
- Adrià Garriga-Alonso and Vincent Fortuin. Exact langevin dynamics with stochastic gradients. *arXiv preprint arXiv:2102.01691*, 2021.
- Ethan Goan and Clinton Fookes. Bayesian neural networks: An introduction and survey. *Case studies in applied Bayesian data science: CIRM Jean-Morlet chair, Fall 2018*, pages 45–87, 2020.
- Zhaohan Guo, Shantanu Thakoor, Miruna Pîslar, Bernardo Avila Pires, Florent Altché, Corentin Tallec, Alaa Saade, Daniele Calandriello, Jean-Bastien Grill, Yunhao Tang, et al. BYOL-Explore: Exploration by bootstrapped prediction. *Advances in neural information processing systems*, 35:31855–31870, 2022.
- Bobby He, Balaji Lakshminarayanan, and Yee Whye Teh. Bayesian deep ensembles via the neural tangent kernel. *Advances in neural information processing systems*, 33, 2020.
- Haqee Ishfaq, Qiwen Cui, Viet Nguyen, Alex Ayoub, Zhuoran Yang, Zhaoran Wang, Doina Precup, and Lin Yang. Randomized exploration in reinforcement learning with general value function approximation. In *International conference on machine learning*, pages 4607–4616. PMLR, 2021.
- Pavel Izmailov, Sharad Vikram, Matthew D Hoffman, and Andrew Gordon Gordon Wilson. What are Bayesian neural network posteriors really like? In *International conference on machine learning*, pages 4629–4640. PMLR, 2021.
- Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

- Mohammad Emtiyaz Khan, Alexander Immer, Ehsan Abedi, and Maciej Korzepa. Approximate inference turns deep networks into gaussian processes. *Advances in neural information processing systems*, 32, 2019.
- Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In Yoshua Bengio and Yann LeCun, editors, *International conference on learning representations*, 2014.
- Seijin Kobayashi, Pau Vilimelis Aceituno, and Johannes Von Oswald. Disentangling the predictive variance of deep ensembles through the neural tangent kernel. *Advances in Neural Information Processing Systems*, 35: 25335–25348, 2022.
- Salem Lahlou, Moksh Jain, Hadi Nekoei, Victor Ion Butoi, Paul Bertin, Jarrid Rector-Brooks, Maksym Korablyov, and Yoshua Bengio. Deup: Direct epistemic uncertainty prediction. *arXiv preprint arXiv:2102.08501*, 2021.
- Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.
- Olivier Laurent, Adrien Lafage, Enzo Tartaglione, Geoffrey Daniel, Jean-Marc Martinez, Andrei Bursuc, and Gianni Franchi. Packed-ensembles for efficient uncertainty estimation. *arXiv preprint arXiv:2210.09184*, 2022.
- Jaehoon Lee, Jascha Sohl-dickstein, Jeffrey Pennington, Roman Novak, Sam Schoenholz, and Yasaman Bahri. Deep neural networks as Gaussian processes. In *International conference on learning representations*, 2018.
- Jaehoon Lee, Samuel Schoenholz, Jeffrey Pennington, Ben Adlam, Lechao Xiao, Roman Novak, and Jascha Sohl-Dickstein. Finite versus infinite neural networks: an empirical study. *Advances in Neural Information Processing Systems*, 33:15156–15172, 2020a.
- Jaehoon Lee, Lechao Xiao, Samuel S. Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide Neural Networks of Any Depth Evolve as Linear Models Under Gradient Descent. *Journal of Statistical Mechanics: Theory and Experiment*, 2020, December 2020b.
- Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose Bayesian inference algorithm. *Advances in neural information processing systems*, 29, 2016.
- Clare Lyle, Mark Rowland, Will Dabney, Marta Kwiatkowska, and Yarin Gal. Learning dynamics and generalization in reinforcement learning. *arXiv preprint arXiv:2206.02126*, 2022.
- Michael Matthews, Michael Beukman, Benjamin Ellis, Mikayel Samvelyan, Matthew Jackson, Samuel Coward, and Jakob Foerster. Craftax: A lightning-fast benchmark for open-ended reinforcement learning. *arXiv preprint arXiv:2402.16801*, 2024.
- Radford M Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, 1996.
- Nikolay Nikolov, Johannes Kirschner, Felix Berkenkamp, and Andreas Krause. Information-directed exploration for deep reinforcement learning. In *International conference on learning representations, ICLR*, 2019.
- Alexander Nikulin, Vladislav Kurenkov, Denis Tarasov, and Sergey Kolesnikov. Anti-exploration by random network distillation. In *International Conference on Machine Learning*, pages 26228–26244. PMLR, 2023.
- Atsushi Nitanda and Taiji Suzuki. Optimal rates for averaged stochastic gradient descent under neural tangent kernel regime. In *International conference on learning representations*, 2021.
- Kazuki Osawa, Siddharth Swaroop, Mohammad Emtiyaz Khan, Anirudh Jain, Runa Eschenhagen, Richard E Turner, and Rio Yokota. Practical deep learning with bayesian principles. *Advances in neural information processing systems*, 32, 2019.
- Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped DQN. *Advances in neural information processing systems*, 29, 2016.
- Ian Osband, Benjamin Van Roy, Daniel J Russo, Zheng Wen, et al. Deep exploration via randomized value functions. *Journal of machine learning research*, 20, 2019.
- Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*. PMLR, 2017.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.
- Tim GJ Rudner, Zonghao Chen, Yee Whye Teh, and Yarin Gal. Tractable function-space variational inference in bayesian neural networks. *Advances in neural information processing systems*, 35:22686–22698, 2022.

- Maxim Samarin, Volker Roth, and David Belius. On the empirical neural tangent kernel of standard finite-width convolutional neural network architectures. *arXiv preprint arXiv:2006.13645*, 2020.
- Mariia Seleznova and Gitta Kutyniok. Analyzing finite neural networks: Can we trust neural tangent kernel theory? In *Mathematical and Scientific Machine Learning*, pages 868–895. PMLR, 2022.
- Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems*, 31, 2018.
- Natasa Tagasovska and David Lopez-Paz. Single-model uncertainties for deep learning. *Advances in neural information processing systems*, 32, 2019.
- Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International conference on machine learning*, pages 9690–9700. PMLR, 2020.
- Hoi-To Wai, Zhuoran Yang, Zhaoran Wang, and Mingyi Hong. Provably efficient neural GTD for off-policy learning. *Advances in Neural Information Processing Systems*, 33:10431–10442, 2020.
- Joseph Wilson, Chris van der Heide, Liam Hodgkinson, and Fred Roosta. Uncertainty quantification with the empirical neural tangent kernel. *arXiv preprint arXiv:2502.02870*, 2025.
- Chenjun Xiao, Bo Dai, Jincheng Mei, Oscar A Ramirez, Ramki Gummadi, Chris Harris, and Dale Schuurmans. Understanding and leveraging overparameterization in recursive value estimation. In *International Conference on Learning Representations*, 2021.
- Greg Yang. Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. *arXiv preprint arXiv:1902.04760*, 2019.
- Zhuoran Yang, Chi Jin, Zhaoran Wang, Mengdi Wang, and Michael Jordan. Provably efficient reinforcement learning with kernel and neural function approximations. *Advances in Neural Information Processing Systems*, 33: 13903–13916, 2020.
- Moritz Akiya Zanger, Wendelin Böhmer, and Matthijs T J Spaan. Diverse projection ensembles for distributional reinforcement learning. In *International conference on learning representations*, 2024.
- Moritz Akiya Zanger, Pascal R. Van der Vaart, Wendelin Boehmer, and Matthijs T. J. Spaan. Contextual similarity distillation: Ensemble uncertainties with a single model. In *International conference on learning representations*, 2026a.
- Moritz Akiya Zanger, Max Weltevrede, Yaniv Oren, Pascal R. Van der Vaart, Caroline Horsch, Wendelin Boehmer, and Matthijs T. J. Spaan. Universal value-function uncertainties. In *International conference on learning representations*, 2026b.

On the Equivalence of Random Network Distillation, Deep Ensembles, and Bayesian Inference

(Supplementary Material)

Moritz A. Zanger¹ Yijun Wu¹ Pascal R. van der Vaart¹ Wendelin Böhmer¹ Matthijs T. J. Spaan¹

¹Delft University of Technology, Delft, 2628 XE, The Netherlands

A LIMITATIONS AND ASSUMPTIONS

We provide an overview of the primary assumptions underpinning our analysis and discuss their relation to practical settings. The foremost assumption is that our analysis operates within the NTK regime. This framework presupposes the asymptotic limit of infinitely wide neural networks and a so-called NTK-parametrization of forward computations that ensures network dynamics linearize around their initialization, leading to “lazy” learning with kernel regression behavior. This idealized setting naturally deviates from practical implementations involving finite-width networks. Nonetheless, a significant body of work has demonstrated that predictions from NTK theory can remain remarkably accurate for sufficiently wide, modern architectures, providing a reasonable approximation of their behavior [e.g., Lee et al., 2020a, Seleznova and Kutyniok, 2022, Samarin et al., 2020].

Furthermore, our derivations assume training via full-batch gradient flow, which corresponds to gradient descent with an infinitesimal step size. This abstains from the use of stochastic minibatch optimizers, which are standard in practice. While beyond our current scope, extensions of NTK analysis to incorporate the effects of stochastic gradient noise do exist [e.g., Yang, 2019, Cao and Gu, 2019, Nitanda and Suzuki, 2021]. Finally, our analysis considers a fixed training dataset $\mathcal{X}$. This contrasts with prominent applications of RND, particularly in online reinforcement learning, where the agent interacts with an environment and learns from an inherently non-stationary data stream. Characterizing how these equivalences with ensembles and Bayesian posteriors evolve under such distribution shifts remains an important open question.

B PROOFS

This section provides extended proofs for our analysis of RND.

B.1 ENSEMBLE EQUIVALENCE

Our first result states the equivalence of self-predictive errors of RND and predictive variance of deep ensembles in the infinite-width NTK regime. For completeness, we also include proofs or simplified proof sketches for known results that support our analysis.

Theorem B.1. *[Jacot et al., 2018](Post-convergence neural network function) In the limit of infinite layer widths $n \rightarrow \infty$ and infinite time $t \rightarrow \infty$, the output function of a neural network $f(x; \theta_\infty)$ with NTK parametrization according to Eq. 1 is given by*

$$f(x; \theta_\infty) = f(x; \theta_0) - \Theta_{x\mathcal{X}} \Theta_{\mathcal{X}\mathcal{X}}^{-1} (\mathcal{Y} - f(\mathcal{X}; \theta_0)),$$

where we used the shorthand $\Theta_{xx'} \equiv \Theta(x, x')$.

Proof sketch. By taking the infinite width limit $n \rightarrow \infty$, we obtain a linear ODE from Eq. (5). Through an exponential ansatz, its explicit solution with initial condition $f(x; \theta_0)$ is given by $f(x; \theta_t) = f(x; \theta_0) + \Theta_{x\mathcal{X}} \Theta_{\mathcal{X}\mathcal{X}}^{-1} (I - e^{-t\Theta_{\mathcal{X}\mathcal{X}}})(\mathcal{Y} - f(\mathcal{X}; \theta_0))$.

Assuming the training Gram matrix $\Theta_{\mathcal{X}\mathcal{X}}$ is positive definite (and thus invertible), the exponential term decays to zero as $t \rightarrow \infty$, yielding the kernel regression formula in Proposition (B.1). See Jacot et al. [2018] and Appendix B.1.1.

B.1.1 Proof of Theorem B.1

Proof. The proof is centered around the learning dynamics of a neural network under gradient descent, whereby we assume the limit of infinitesimal step size for simplicity. This setting is also referred to as “gradient flow”. The driving force behind the learning dynamics of parameters θ_t is gradient flow optimization on the loss

$$\mathcal{L}(\theta_t) = \frac{1}{2} \|f(\mathcal{X}, \theta_t) - \mathcal{Y}\|_2^2, \quad (15)$$

with the subsequent evolution of parameters by

$$\frac{d}{dt}\theta_t = -\alpha \nabla_{\theta} \mathcal{L}(\theta_t), \quad (16)$$

where α is a learning rate. From this, we can obtain the parameter space differential equation

$$\frac{d}{dt}\theta_t = -\alpha \nabla_{\theta} f(\mathcal{X}, \theta_t) (f(\mathcal{X}, \theta_t) - \mathcal{Y}). \quad (17)$$

In order to translate this expression to a function-space view through a first-order Taylor expansion of f around its initialization parameters θ_0 :

$$f_{\text{lin}}(x, \theta_t) = f(x, \theta_0) + \nabla_{\theta} f(x, \theta_0)^{\top} (\theta_t - \theta_0). \quad (18)$$

The use of a linearized neural network function simplifies the analysis in two aspects: 1.) the linearization offers a simple translation of the parameter space evolution $\frac{d}{dt}\theta_t$ to a function-space evolution and 2.) the linearized neural network function $f_{\text{lin}}(x, \theta_t)$ results in linear dynamics, simplifying the earlier derived differential equation to a linear ODE. The evolution of f_{lin} is then obtained by taking the time-derivative of Eq. (18) and plugging in the parameter evolution for a linearized function from Eq. (17) such that

$$\frac{d}{dt}f_{\text{lin}}(x, \theta_t) = -\alpha \nabla_{\theta} f(x, \theta_0)^{\top} \nabla_{\theta} f(\mathcal{X}, \theta_0) (f_{\text{lin}}(\mathcal{X}, \theta_t) - \mathcal{Y}). \quad (19)$$

Let us denote the training error of f_{lin} at time t with $\delta_t = f_{\text{lin}}(\mathcal{X}, \theta_t) - \mathcal{Y}$ and accordingly write

$$\frac{d}{dt}\delta_t = -\alpha \Theta_{\mathcal{X}\mathcal{X}}^0 \delta_t, \quad (20)$$

where $\Theta_{\mathcal{X}\mathcal{X}}^0$ denotes the empirical tangent kernel $\Theta_{\mathcal{X}\mathcal{X}}^0 = \nabla_{\theta} f(\mathcal{X}, \theta_0)^{\top} \nabla_{\theta} f(\mathcal{X}, \theta_0)$ at initialization. The differential equation (20) is a linear ODE system to which an exponential ansatz provides the explicit solution

$$\delta_t = e^{-\alpha t \Theta_{\mathcal{X}\mathcal{X}}^0} \delta_0, \quad (21)$$

where $e^{\Theta_{\mathcal{X}\mathcal{X}}} = \sum_{k=0}^{\infty} \frac{1}{k!} (\Theta_{\mathcal{X}\mathcal{X}})^k$ is the matrix exponential. We plug this result back in the linearized function space differential equation 19 to obtain

$$\frac{d}{dt}f_{\text{lin}}(x, \theta_t) = -\alpha \Theta_{x\mathcal{X}}^0 e^{-\alpha t \Theta_{\mathcal{X}\mathcal{X}}^0} (f(\mathcal{X}, \theta_0) - \mathcal{Y}). \quad (22)$$

In this form, we can solve for $f_{\text{lin}}(x, \theta_t)$ directly by integration

$$f_{\text{lin}}(x, \theta_t) = f(x, \theta_0) + \int_0^t \frac{d}{dt'} f_{\text{lin}}(x, \theta_{t'}) dt' \quad (23)$$

$$= f(x, \theta_0) + \Theta_{x\mathcal{X}}^0 (\Theta_{\mathcal{X}\mathcal{X}}^0)^{-1} \left(e^{-\alpha t \Theta_{\mathcal{X}\mathcal{X}}^0} - I \right) (f(\mathcal{X}, \theta_0) - \mathcal{Y}). \quad (24)$$

Remarkably, the linearized and true learning dynamics become increasingly aligned with increasing neural network width. Jacot et al. [2018] and Lee et al. [2020b] show that as network width increases, the required individual movement of parameters $\theta_t - \theta_0$ to effect sufficient movement in the output function $f(x, \theta_t)$ decreases. In the limit of infinite width $n \rightarrow \infty$, the linearization of f then becomes exact $\lim_{n \rightarrow \infty} f_{\text{lin}}(x, \theta_t) = f(x, \theta_t)$. Under the outlined training dynamics, the same limit furthermore causes the NTK to become deterministic (despite random weight initializations) and stationary $\lim_{n \rightarrow \infty} \Theta_{xx'}^0 = \Theta_{xx'}^t = \Theta_{xx'}$. Thus, the converged function at time $t \rightarrow \infty$ is described by

$$f(x, \theta_{\infty}) = f(x, \theta_0) - \Theta_{x\mathcal{X}} \Theta_{\mathcal{X}\mathcal{X}}^{-1} (f(\mathcal{X}, \theta_0) - \mathcal{Y}). \quad (25)$$

□

B.1.2 Proof of Theorem 2.1

We restate Theorem 2.1 for convenience.

Theorem 2.1. [Lee et al., 2020b](Distribution of post-convergence neural network functions) Let $f(\mathcal{X}_T; \theta_\infty)$ be a NN as defined in Eq.(1), and let $\mathcal{X}_T$ be testpoints. For random initializations $\theta_0 \sim \mathcal{N}(0, I)$, and in the limit $n \rightarrow \infty$, $f(\mathcal{X}_T; \theta_\infty)$ distributes as a Gaussian with mean and covariance given by

$$\begin{aligned}\mathbb{E}[f(\mathcal{X}_T, \theta_\infty)] &= \Theta_{\mathcal{X}_T \mathcal{X}} \Theta_{\mathcal{X} \mathcal{X}}^{-1} \mathcal{Y}, \\ \Sigma_{\mathcal{X}_T \mathcal{X}_T}^f(\theta_\infty) &= \kappa_{\mathcal{X}_T \mathcal{X}_T} + \Theta_{\mathcal{X}_T \mathcal{X}} \Theta_{\mathcal{X} \mathcal{X}}^{-1} \kappa_{\mathcal{X} \mathcal{X}} \Theta_{\mathcal{X} \mathcal{X}}^{-1} \Theta_{\mathcal{X} \mathcal{X}_T} \\ &\quad - (\Theta_{\mathcal{X}_T \mathcal{X}} \Theta_{\mathcal{X} \mathcal{X}}^{-1} \kappa_{\mathcal{X} \mathcal{X}_T} + \text{h.c.}),\end{aligned}$$

where h.c. is the Hermitian conjugate of the preceding term.

Proof sketch. We use the fact that $f(x; \theta_\infty)$ can be written as a linear combination of the test initialization $f(x; \theta_0)$ and the training initialization $f(\mathcal{X}; \theta_0)$. Both these identities are described probabilistically by the NNGP $f(x; \theta_0) \sim \mathcal{GP}(0, \kappa_{xx'})$, and $f(\mathcal{X}; \theta_0) \sim \mathcal{GP}(0, \kappa_{\mathcal{X} \mathcal{X}})$. Applying a linear transformation to a GP yields another GP [Rasmussen and Williams, 2006], meaning $f(x; \theta_\infty)$ also follows a GP. Propagating the prior covariance κ through the linear transformation described by Proposition B.1 reveals the expression for the post-convergence covariance function $\Sigma_{\mathcal{X}_T \mathcal{X}_T}^f(\theta_\infty)$ given in Theorem 2.1.

Proof. The proof builds on the previous result of Proposition B.1 providing a closed-form expression for the post-convergence function as a deterministic function of its initialization, here evaluated for a set of test points $\mathcal{X}_T$

$$f(\mathcal{X}_T, \theta_\infty) = f(\mathcal{X}_T, \theta_0) - \Theta_{\mathcal{X}_T \mathcal{X}} \Theta_{\mathcal{X} \mathcal{X}}^{-1} (f(\mathcal{X}, \theta_0) - \mathcal{Y}). \quad (26)$$

To be precise, the post-convergence predictions $f(\mathcal{X}_T, \theta_\infty)$ can be written as an affine transformation of the vector $(f(\mathcal{X}_T, \theta_0), f(\mathcal{X}, \theta_0)^\top)^\top$. This yields the block matrix equation

$$\begin{pmatrix} f(\mathcal{X}_T, \theta_\infty) \\ f(\mathcal{X}, \theta_\infty) \end{pmatrix} = \begin{pmatrix} I & -\Theta(\mathcal{X}_T, \mathcal{X}) \Theta(\mathcal{X}, \mathcal{X})^{-1} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} f(\mathcal{X}_T, \theta_0) \\ f(\mathcal{X}, \theta_0) \end{pmatrix} + \begin{pmatrix} \Theta(\mathcal{X}_T, \mathcal{X}) \Theta(\mathcal{X}, \mathcal{X})^{-1} \mathcal{Y} \\ \mathcal{Y} \end{pmatrix}. \quad (27)$$

We recall that, at initialization, neural networks in the infinite width limit distribute to a GP called NNGP [Lee et al., 2018] as

$$f(\mathcal{X}_T, \theta_0) \sim \mathcal{GP}(0, \kappa_{\mathcal{X}_T \mathcal{X}_T}) \quad \text{where} \quad \kappa_{\mathcal{X}_T \mathcal{X}_T} = \mathbb{E}_{\theta_0}[f(\mathcal{X}_T, \theta_0) f(\mathcal{X}_T, \theta_0)^\top]. \quad (28)$$

The block eq. (27) thus describes an affine transformation of a GP itself. We have that affine transformations of multivariate Gaussian random variables $X \sim \mathcal{N}(\mu_X, \Sigma_X)$ with $Y = a + BX$ distribute Gaussian themselves with $Y \sim \mathcal{N}(a + B\mu_X, B\Sigma_X B^\top)$. Application to Eq. 27 and rearrangement then yields the post-convergence GP with mean and covariance

$$\mathbb{E}[f(\mathcal{X}_T, \theta_\infty)] = \Theta_{\mathcal{X}_T \mathcal{X}} \Theta_{\mathcal{X} \mathcal{X}}^{-1} \mathcal{Y}, \quad (29)$$

$$\begin{aligned}\Sigma_{\mathcal{X}_T \mathcal{X}_T}^f(\theta_\infty) &= \\ &\kappa_{\mathcal{X}_T \mathcal{X}_T} + \Theta_{\mathcal{X}_T \mathcal{X}} \Theta_{\mathcal{X} \mathcal{X}}^{-1} \kappa_{\mathcal{X} \mathcal{X}} \Theta_{\mathcal{X} \mathcal{X}}^{-1} \Theta_{\mathcal{X} \mathcal{X}_T} - (\Theta_{\mathcal{X}_T \mathcal{X}} \Theta_{\mathcal{X} \mathcal{X}}^{-1} \kappa_{\mathcal{X} \mathcal{X}_T} + \text{h.c.}),\end{aligned} \quad (30)$$

where h.c. refers to the Hermitian conjugate of the preceding term. This completes the proof. $\square$

B.1.3 Proof of Theorem 3.1

We restate Theorem 3.1 for convenience.

Theorem 3.1. (Distribution of post-convergence RND errors) Under NTK parametrization, let $u(x; \vartheta_\infty)$ be a converged prediction network in $t \rightarrow \infty$, with data $\mathcal{X}$ and fixed target network $g(\mathcal{X}; \psi_0)$. Let parameters ϑ_0, ψ_0 be drawn i.i.d.

$\vartheta_0, \psi_0 \sim \mathcal{N}(0, I)$, with the resulting NNGP $u(x; \vartheta_0) \sim \mathcal{GP}(0, \kappa^u(x, x'))$ and $g(x; \psi_0) \sim \mathcal{GP}(0, \kappa^g(x, x'))$. The post-convergence RND error $\epsilon(\mathcal{X}_T; \vartheta_\infty, \psi_0)$ is Gaussian with zero mean and covariance

$$\begin{aligned} \mathbb{E}[\epsilon(\mathcal{X}_T, \vartheta_\infty, \psi_0)] &= 0, \\ \Sigma_{\mathcal{X}_T \mathcal{X}_T}^\epsilon(\vartheta_\infty, \psi_0) &= \kappa_{\mathcal{X}_T \mathcal{X}_T}^\epsilon + \Theta_{\mathcal{X}_T \mathcal{X}} \Theta_{\mathcal{X} \mathcal{X}}^{-1} \kappa_{\mathcal{X} \mathcal{X}}^\epsilon \Theta_{\mathcal{X} \mathcal{X}}^{-1} \Theta_{\mathcal{X} \mathcal{X}_T} \\ &\quad - (\Theta_{\mathcal{X}_T \mathcal{X}} \Theta_{\mathcal{X} \mathcal{X}}^{-1} \kappa_{\mathcal{X} \mathcal{X}_T}^\epsilon + h.c.), \end{aligned}$$

where $\kappa_{xx'}^\epsilon = \kappa_{xx'}^u + \kappa_{xx'}^g$, is the covariance kernel of initialization errors $\epsilon(x; \vartheta_0, \psi_0) = u(x; \vartheta_0) - g(x; \psi_0)$.

Proof. This proposition considers the post-convergence distribution of self-predictive errors as produced by RND. The online predictor $u(x; \vartheta_t)$ undergoes learning dynamics under the same conditions as outlined in the derivation of Proposition B.1, albeit with the self-predictive loss

$$\mathcal{L}(\vartheta_t) = \frac{1}{2} \|u(\mathcal{X}, \vartheta_t) - g(\mathcal{X}, \psi_0)\|_2^2. \quad (31)$$

This, by analogy to Theorem B.1, implies that the online predictor $u(x; \vartheta_t)$ converges as $t \rightarrow \infty$ to the function

$$u(x, \vartheta_\infty) = u(x, \vartheta_0) - \Theta_{x \mathcal{X}} \Theta_{\mathcal{X} \mathcal{X}}^{-1} (u(\mathcal{X}, \vartheta_0) - g(\mathcal{X}, \psi_0)). \quad (32)$$

For a set of test points $\mathcal{X}_T$, the error $\epsilon(\mathcal{X}_T; \vartheta_\infty, \psi_0) = u(\mathcal{X}_T; \vartheta_\infty) - g(\mathcal{X}_T; \psi_0)$ at convergence can thus be written as the affine transformation

$$\epsilon(\mathcal{X}_T; \vartheta_\infty, \psi_0) = \epsilon(\mathcal{X}_T; \vartheta_0, \psi_0) - \Theta_{\mathcal{X}_T \mathcal{X}} \Theta_{\mathcal{X} \mathcal{X}}^{-1} \epsilon(\mathcal{X}; \vartheta_0, \psi_0). \quad (33)$$

and the corresponding block matrix equation

$$\begin{pmatrix} \epsilon(\mathcal{X}_T; \vartheta_\infty, \psi_0) \\ \epsilon(\mathcal{X}; \vartheta_\infty, \psi_0) \end{pmatrix} = \begin{pmatrix} I & -\Theta_{\mathcal{X}_T \mathcal{X}} \Theta_{\mathcal{X} \mathcal{X}}^{-1} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \epsilon(\mathcal{X}_T; \vartheta_0, \psi_0) \\ \epsilon(\mathcal{X}; \vartheta_0, \psi_0) \end{pmatrix}. \quad (34)$$

The errors accordingly are themselves Gaussian with $\epsilon(\mathcal{X}_T; \vartheta_\infty, \psi_0) \sim \mathcal{GP}(0, \kappa_{\mathcal{X}_T \mathcal{X}_T}^\epsilon)$ where $\kappa_{\mathcal{X}_T \mathcal{X}_T}^\epsilon = \mathbb{E}_{\vartheta_0, \psi_0}[\epsilon(\mathcal{X}_T; \vartheta_0, \psi_0) \epsilon(\mathcal{X}_T; \vartheta_0, \psi_0)^\top]$. The latter term describes the distribution of self-predictive errors at initialization, which is a simple sum of two independent NNGP $\epsilon(\mathcal{X}_T; \vartheta_0, \psi_0) = u(\mathcal{X}_T; \vartheta_0) - g(\mathcal{X}_T; \psi_0)$ such that $\kappa_{\mathcal{X}_T \mathcal{X}_T}^\epsilon = \kappa_{\mathcal{X}_T \mathcal{X}_T}^u + \kappa_{\mathcal{X}_T \mathcal{X}_T}^g$, completing the proof. $\square$

B.1.4 Proof of Proposition 3.3

Before treating Proposition 3.3 we first derive two known results concerning the independence and recursive character of the NNGP kernel and the NTK. We assume forward computations of $f(x; \theta_t)$ are defined according to Eq. 1. To avoid confusion with indices i, j we will in this section use the notation $\kappa(x, x')$ rather than $\kappa_{xx'}$ to denote the function inputs x, x' (and similarly for $\Theta(x, x')$).

Proposition B.2. [Lee et al., 2018] (Recursive NNGP formulation) At initialization $t = 0$ and in the limit $n \rightarrow \infty$, the i -th output at layer l , $z_i^l(x; \theta_0^{\leq l})$, converges to a GP with zero mean and covariance function $\kappa_{ii}^l(x, x')$ given by

$$\kappa_{ii}^1(x, x') = \frac{\sigma_w^2}{n_0} x^\top x' + \sigma_b^2, \quad \text{and} \quad \kappa_{ij}^1(x, x') = 0, \quad \text{if } i \neq j, \quad (35)$$

$$\kappa_{ii}^l(x, x') = \sigma_b^2 + \sigma_w^2 \mathbb{E}_{z_i^{l-1} \sim \mathcal{GP}(0, \kappa_{ii}^{l-1})} [\phi(z_i^{l-1}(x; \theta_0^{\leq l-1})) \phi(z_i^{l-1}(x'; \theta_0^{\leq l-1}))], \quad (36)$$

$$\text{and} \quad \kappa_{ij}^l(x, x') = 0, \quad \text{if } i \neq j, \quad (37)$$

and we have $\kappa_{ii}^l(x, x') = \kappa^l(x, x')$, $\forall i$.

Proof. We prove the proposition by induction. The induction assumption is that if outputs at layer $l-1$ satisfy a GP structure

$$z_i^{l-1} \sim \mathcal{GP}(0, \kappa^{l-1}), \quad (38)$$

with the covariance function defined as

$$\kappa_{ij}^{l-1}(x, x') = \mathbb{E}[z_i^{l-1}(x; \theta_0^{\leq l-1}) z_j^{l-1}(x'; \theta_0^{\leq l-1})] = \begin{cases} k^{l-1}(x, x') & \text{if } i = j, \\ 0 & \text{if } i \neq j, \end{cases} \quad (39)$$

then, outputs at layer l follow

$$z_i^l(x) \sim \mathcal{GP}(0, \kappa^l), \quad (40)$$

where the NNGP kernel at layer l is given by:

$$\kappa_{ii}^l(x, x') = \mathbb{E}[z_i^l(x; \theta_0^{\leq l}) z_i^l(x'; \theta_0^{\leq l})] = \kappa^l(x, x'), \quad \forall i, \quad (41)$$

$$\kappa_{ij}^l(x, x') = \mathbb{E}[z_i^l(x; \theta_0^{\leq l}) z_j^l(x'; \theta_0^{\leq l})] = 0, \quad \text{if } i \neq j. \quad (42)$$

with the recursive definition

$$\kappa^l(x, x') = \sigma_b^2 + \sigma_w^2 \mathbb{E}_{z_i^{l-1} \sim \mathcal{GP}(0, \kappa^{l-1})} [\phi(z_i^{l-1}(x; \theta_0^{\leq l-1})) \phi(z_i^{l-1}(x'; \theta_0^{\leq l-1}))]. \quad (43)$$

Base case ($l = 1$). At layer $l = 1$ we have:

$$z_i^1(x; \theta_0^{\leq 1}) = \frac{\sigma_w}{\sqrt{n_0}} \sum_{j=1}^{n_0} w_{ij}^1 x_j + \sigma_b b_i^1. \quad (44)$$

This is an affine transform of Gaussian random variables; thus, $z_i^1(x; \theta_0^{\leq 1})$ distributes Gaussian with

$$z_i^1(x) \sim \mathcal{GP}(0, \kappa^1), \quad (45)$$

with kernel

$$\kappa^1(x, x') = \frac{\sigma_w^2}{n_0} x^\top x' + \sigma_b^2 = \kappa_{ii}^1(x, x'), \quad \text{and} \quad \kappa_{ij}^1 = 0, \quad \text{if } i \neq j, \quad (46)$$

where the independence follows from the fact that $z_i^1(x; \theta_0^{\leq 1})$ is computed from separate, independent rows of weights and biases.

Induction step $l > 1$. For layers $l > 1$ we have

$$z_i^l(x; \theta_0^{\leq l}) = \sigma_b b_i^l + \frac{\sigma_w}{\sqrt{n_{l-1}}} \sum_{j=1}^{n_{l-1}} w_{ij}^l x_j^l(x), \quad x_j^l(x) = \phi(z_j^{l-1}(x; \theta_0^{\leq l-1})). \quad (47)$$

By the induction assumption, $z_j^{l-1}(x; \theta_0^{\leq l-1})$ are generated by independent GP. Hence, $x_i^l(x)$ and $x_j^l(x)$ are independent for $i \neq j$. Consequently, $z_i^l(x; \theta_0^{\leq l})$ is a sum of independent random variables. By the CLT (as $n_1, \dots, n_L \rightarrow \infty$) the tuple $\{z_i^l(x; \theta_0^{\leq l}), z_i^l(x'; \theta_0^{\leq l})\}$ tends to be jointly Gaussian, with covariance given by:

$$\begin{aligned} \mathbb{E}[z_i^l(x; \theta_0^{\leq l}) z_i^l(x'; \theta_0^{\leq l})] &= \\ \sigma_b^2 + \sigma_w^2 \mathbb{E}_{z_i^{l-1} \sim \mathcal{GP}(0, \kappa^{l-1})} [\phi(z_i^{l-1}(x; \theta_0^{\leq l-1})) \phi(z_i^{l-1}(x'; \theta_0^{\leq l-1}))]. \end{aligned} \quad (48)$$

Moreover, as z_i^l and z_j^l for $i \neq j$ are defined through independent rows of the parameters w^l, b^l and independent pre-activations $x^l(x)$, we have

$$\kappa_{ij}^l = \mathbb{E}[z_i^l(x) z_j^l(x')] = 0, \quad \text{if } i \neq j, \quad (49)$$

and thus completing the proof. $\square$

Proposition B.3. [Jacot et al., 2018] (Recursive NTK formulation) In the limit $n \rightarrow \infty$, the neural tangent kernel $\Theta_{ii}^l(x, x')$ of the i -th output $z_i^l(x; \theta_0^{\leq l})$ at layer l , defined as the gradient inner product

$$\Theta_{ii}^l(x, x') = \nabla_{\theta^l} z_i^l(x; \theta_0^{\leq l})^\top \nabla_{\theta^l} z_i^l(x'; \theta_0^{\leq l}), \quad (50)$$

is given recursively by

$$\Theta_{ii}^1(x, x') = \kappa_{ii}^1(x, x') = \frac{\sigma_w^2}{n_0} x^\top x' + \sigma_b^2, \quad \text{and} \quad \Theta_{ij}^1(x, x') = 0, \quad \text{if } i \neq j, \quad (51)$$

$$\Theta_{ii}^l(x, x') = \Theta_{ii}^{l-1}(x, x') \dot{\kappa}_{ii}^{l-1}(x, x') + \kappa_{ii}^l(x, x'), \quad (52)$$

$$\Theta_{ij}^l(x, x') = 0 \quad \text{if } i \neq j, \quad (53)$$

where

$$\dot{\kappa}_{ii}^l(x, x') = \sigma_w^2 \mathbb{E}_{z_i^{l-1} \sim \mathcal{GP}(0, \kappa_{ii}^{l-1})} [\dot{\phi}(z_i^{l-1}(x; \theta_0^{\leq l-1})) \dot{\phi}(z_i^{l-1}(x'; \theta_0^{\leq l-1}))], \quad (54)$$

and

$$\Theta_{ij}^l(x, x') = \nabla_{\theta^l} z_i^l(x; \theta_0^{\leq l})^\top \nabla_{\theta^l} z_j^l(x'; \theta_0^{\leq l}) = 0 \quad \text{if } i \neq j. \quad (55)$$

Proof. The proof is by induction. The induction assumption is that if gradients satisfy at layer $l-1$

$$\begin{aligned} \Theta_{ij}^{l-1}(x, x') &= \\ \nabla_{\theta^{l-1}} z_i^{l-1}(x; \theta_0^{\leq l-1})^\top \nabla_{\theta^{l-1}} z_j^{l-1}(x'; \theta_0^{\leq l-1}) &= \begin{cases} \Theta_{ii}^{l-1}(x, x') & \text{if } i = j, \\ 0 & \text{if } i \neq j, \end{cases} \end{aligned} \quad (56)$$

then at layer l we have

$$\Theta_{ij}^l(x, x') = \begin{cases} \Theta_{ii}^{l-1}(x, x') \dot{\kappa}_{ii}^{l-1}(x, x') + \kappa_{ii}^l(x, x') & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases} \quad (57)$$

Base case ($l = 1$). At layer $l = 1$, we have

$$z_i^1(x; \theta_0^{\leq 1}) = \sigma_b b_i^1 + \frac{\sigma_w}{\sqrt{n_0}} \sum_j^{n_0} w_{ij}^1 x_j, \quad (58)$$

and the gradient inner product is given by:

$$\nabla_{\theta^1} z_i^1(x; \theta_0^{\leq 1})^\top \nabla_{\theta^1} z_i^1(x'; \theta_0^{\leq 1}) = \frac{\sigma_w^2}{n_0} x^\top x' + \sigma_b^2 = \kappa_{ii}^1(x, x'). \quad (59)$$

Inductive step ($l > 1$). For layers $l > 1$, we split parameters $\theta^l = \theta^{l-1} \cup \{w^l, b^l\}$ and split the inner product by

$$\Theta_{ii}^l(x, x') = \underbrace{\nabla_{\theta^{l-1}} z_i^{l-1}(x; \theta_0^{\leq l-1})^\top \nabla_{\theta^{l-1}} z_i^{l-1}(x'; \theta_0^{\leq l-1})}_{l.h.s.} + \underbrace{\nabla_{\{w^l, b^l\}} z_i^l(x; \theta_0^{\leq l})^\top \nabla_{\{w^l, b^l\}} z_i^l(x'; \theta_0^{\leq l})}_{r.h.s.}. \quad (60)$$

Note that the above *r.h.s* involves gradients w.r.t. last-layer parameters, i.e. the post-activation outputs of the previous layer, and by the same arguments as in the NNGP derivation of Proposition B.2, this is a sum of independent post activations s.t. in the limit $n_{l-1} \rightarrow \infty$

$$\nabla_{\{w^l, b^l\}} z_i^l(x; \theta_0^{\leq l})^\top \nabla_{\{w^l, b^l\}} z_j^l(x'; \theta_0^{\leq l}) = \begin{cases} \kappa_{ii}^l(x, x'), & i = j, \\ 0, & i \neq j. \end{cases} \quad (61)$$

For the *l.h.s.*, we first apply chain rule to obtain

$$\nabla_{\theta^{l-1}} z_i^{l-1}(x; \theta_0^{\leq l-1}) = \frac{\sigma_w}{\sqrt{n_{l-1}}} \sum_j^{n_{l-1}} w_{ij}^l \dot{\phi}(z_j^{l-1}(x; \theta_0^{\leq l-1})) \nabla_{\theta^{l-1}} z_j^{l-1}(x; \theta_0^{\leq l-1}). \quad (62)$$

The gradient inner product of outputs i and j thus reduces to

$$\begin{aligned} \nabla_{\theta^{l-1}} z_i^l(x; \theta_0^{\leq l})^\top \nabla_{\theta^{l-1}} z_j^l(x'; \theta_0^{\leq l}) &= \\ \frac{\sigma_w^2}{n_{l-1}} \sum_k^{n_{l-1}} w_{ik}^l w_{jk}^l \dot{\phi}(z_k^{l-1}(x; \theta_0^{\leq l-1})) \dot{\phi}(z_k^{l-1}(x'; \theta_0^{\leq l-1})) \Theta_{kk}^{l-1}(x, x'). \end{aligned} \quad (63)$$

By the induction assumption $\Theta_{kk}^{l-1}(x, x') = \Theta^{l-1}(x, x')$ and again by the independence of the rows w_i^l and w_j^l for $i \neq j$, the above expression converges in the limit $n_{l-1} \rightarrow \infty$ to an expectation with

$$\Theta_{ij}^l(x, x') = \begin{cases} \Theta^{l-1}(x, x') \kappa_{ii}^l(x, x') + \kappa_{ii}^l(x, x') & i = j, \\ 0 & i \neq j, \end{cases} \quad (64)$$

thereby completing the proof. $\square$

We now restate Proposition 3.3 for convenience.

Proposition 3.3. (*Independence of NN functions*) Under NTK parametrization and in the limit $n \rightarrow \infty$, the random functions $f_i(x; \theta_t)$ of a NN with K output dimensions and shared hidden layers are mutually independent with covariance

$$\Sigma_{xx'}^{ij}(\theta_t) = \mathbb{E}[f_i(x; \theta_t) f_j(x'; \theta_t)] = \begin{cases} \Sigma_{xx'}^f(\theta_t) & i = j, \\ 0 & i \neq j, \end{cases}$$

on the interval $t \in [0, \infty)$.

Proof. We begin by deriving the training dynamics for the output $f_i(x; \theta_t)$ analogously to the proof of Proposition B.1. We denote by $\mathcal{Y}_i$ the labels used to train the function $f_i(x; \theta_t)$. By Proposition B.3, the training dynamics of $f_i(x; \theta_t)$ and $f_j(x; \theta_t)$ are decoupled for $i \neq j$ and we can thus derive Eq. 23 analogously for individual output heads i . Taking the infinite width limit, we obtain at time t

$$f_i(x; \theta_t) = f_i(x; \theta_0) + \Theta_{ii}(x, \mathcal{X}) \Theta_{ii}(\mathcal{X}, \mathcal{X})^{-1} \left(e^{-\alpha t \Theta_{ii}(\mathcal{X}, \mathcal{X})} - I \right) (f_i(\mathcal{X}; \theta_0) - \mathcal{Y}_i). \quad (65)$$

Thus, the output head $f_i(x; \theta_t)$ at time t is a deterministic function of its own initialization only, which itself is characterized by a GP $f_i(x; \theta_0) \sim \mathcal{GP}(0, \kappa_{ii}(x, x'))$ that is independent of output heads $j \neq i$ by Proposition B.2. And thus, since $f_i(x; \theta_t)$ is an affine transform of its own independent initialization terms $f_i(x; \theta_0)$ and $f_i(\mathcal{X}; \theta_0)$, it too must follow an independent GP with $\mathbb{E}_{\theta_0}[f_i(x; \theta_t) f_i(x'; \theta_t)] = \Sigma(x, x'; \theta_t)$ and in particular $\mathbb{E}_{\theta_0}[f_i(x; \theta_t) f_j(x'; \theta_t)] = 0$ if $i \neq j$. $\square$

B.1.5 Proof of Theorem 3.4

We restate Theorem 3.4 for convenience.

Theorem 3.4. (*Distributional equivalence between multi-headed RND and finite deep ensembles*) Under the conditions of Theorem 3.1, let $u_i(x; \vartheta_\infty), g_i(x; \psi_0)$ be the i -th output of predictor and target networks respectively with K output dimensions. Denote their sample mean RND error $\bar{\epsilon}^2(x; \vartheta_\infty, \psi_0) = \frac{1}{K} \sum_{i=1}^K \epsilon_i^2(x; \vartheta_\infty, \psi_0)$. Moreover, let $\{f(x; \theta_\infty^i)\}_{i=1}^{K+1}$ be an ensemble of $K+1$ NNs from i.i.d. initial draws θ_0 . Denote its sample variance $\bar{\sigma}_f^2(x; \theta_\infty^{i \dots K+1}) = \frac{1}{K} \sum_{i=1}^{K+1} (f(x; \theta_\infty^i) - \frac{1}{K+1} \sum_{j=1}^{K+1} f(x; \theta_\infty^j))^2$. The sample mean RND error and sample ensemble variance distribute to the same law

$$\frac{1}{2} \bar{\epsilon}^2(x; \vartheta_\infty, \psi_0) \stackrel{D}{=} \bar{\sigma}_f^2(x; \theta_\infty^{i \dots K+1}), \quad (9)$$

where $\stackrel{D}{=}$ indicates an equality in distribution, namely by a scaled Chi-squared distribution $\bar{\sigma}_f^2(x; \theta_\infty^{i \dots K+1}) \sim \frac{\Sigma_{xx}^f(\theta_\infty)}{K} \chi^2(K)$ with scale $\Sigma_{xx}^f(\theta_\infty)$ given by the analytical variance as given in Theorem 2.1.

Proof. The proof follows by combining the results of Propositions (3.1) and (3.3). We define a multiheaded RND predictor with K output heads $\{u_i(x, \vartheta_t)\}_{i=1}^K$ and a fixed multiheaded target network $\{g_i(x; \psi_0)\}_{i=1}^K$ of equivalent architecture as u_i (i.e., both corresponding to the same NTK Θ) with the corresponding prediction errors $\{\epsilon_i(x; \vartheta_t, \psi_0)\}_{i=1}^K$ accordingly. Let $u_i(x, \vartheta_t)$ be trained such that each head i is trained to match the i -th target output $g_i(x; \psi_0)$.

By Proposition 3.3, the predictions of online predictor heads $\{u_i(x, \vartheta_t)\}_{i=1}^K$ at time t and fixed target networks $\{g_i(x; \psi_0)\}_{i=1}^K$ are each mutually independent with

$$\mathbb{E}_{\vartheta_0}[u_i(x; \vartheta_t)u_j(x; \vartheta_t)] = 0, \quad \text{if } i \neq j, \quad (66)$$

and

$$\mathbb{E}_{\psi_0}[g_i(x; \psi_0)g_j(x; \psi_0)] = 0, \quad \text{if } i \neq j. \quad (67)$$

As a consequence, we also have that

$$\mathbb{E}_{\vartheta_0, \psi_0}[\epsilon_i(x; \vartheta_t, \psi_0)\epsilon_j(x; \vartheta_t, \psi_0)] = 0, \quad \text{if } i \neq j. \quad (68)$$

As previously established in the proof of Proposition 3.3, the multi-headed functions $\{\epsilon_i(x; \vartheta_t, \psi_0)\}_{i=1}^K$ follow equivalent learning dynamics as their scalar-output counterparts. The post-convergence distribution of individual heads $\epsilon_i(x; \vartheta_\infty, \psi_0)$ must therefore equal the scalar-output post-convergence distribution established in Theorem 3.1. Consequently, the errors $\{\epsilon_i(x; \vartheta_t, \psi_0)\}_{i=1}^K$ are independent and identically distributed draws from a Gaussian with mean and covariance

$$\begin{aligned} \mathbb{E}[\epsilon(x, \vartheta_\infty, \psi_0)] &= 0, \\ \Sigma_{xx'}^\epsilon(\vartheta_\infty, \psi_0) &= \kappa_{xx'}^\epsilon + \Theta_{x\mathcal{X}} \Theta_{\mathcal{X}\mathcal{X}}^{-1} \kappa_{\mathcal{X}\mathcal{X}}^\epsilon \Theta_{\mathcal{X}\mathcal{X}}^{-1} \Theta_{\mathcal{X}x'} - (\Theta_{x\mathcal{X}} \Theta_{\mathcal{X}\mathcal{X}}^{-1} \kappa_{\mathcal{X}x'}^\epsilon + \text{h.c.}), \end{aligned}$$

where $\kappa_{xx'}^\epsilon = \kappa_{xx'}^u + \kappa_{xx'}^g$. The sample mean square $\frac{1}{2}\bar{\epsilon}^2(x; \vartheta_\infty, \psi_0) = \frac{1}{2K} \sum_{i=1}^K \epsilon_i^2(x; \vartheta_\infty, \psi_0)$ is then known to follow a scaled Chi-squared distribution with K degrees of freedom

$$\frac{1}{2}\bar{\epsilon}^2(x; \vartheta_\infty, \psi_0) \sim \frac{\frac{1}{2}\Sigma_{xx}^\epsilon(\vartheta_\infty, \psi_0)}{K} \chi^2(K) \quad (69)$$

where $\Sigma_{xx}^\epsilon(\vartheta_\infty, \psi_0)$ is the variance of the GP described in Theorem 3.1.

Conversely, a set of $K+1$ independent neural networks arranged to a deep ensemble $\{f(x; \theta_\infty^i)\}_{i=1}^{K+1}$ in the infinite width limit $n \rightarrow \infty$ and at convergence $t \rightarrow \infty$ are by definition i.i.d. samples from the GP described in Theorem 2.1. As before, the empirical variance defined as $\bar{\sigma}_f^2(x; \theta_\infty^{1 \dots K+1}) = \frac{1}{K} \sum_{i=1}^{K+1} (f(x; \theta_\infty^i) - \frac{1}{K+1} \sum_{j=1}^{K+1} f(x; \theta_\infty^j))^2$ distributes as a scaled Chi-squared distribution with K degrees of freedom

$$\bar{\sigma}_f^2(x; \theta_\infty^{1 \dots K+1}) \sim \frac{\Sigma_{xx}^f(\theta_\infty)}{K} \chi^2(K), \quad (70)$$

where $\Sigma_{xx}^f(\theta_\infty)$ is the variance of the GP described in Theorem 2.1.

Finally, as we assume equal architecture and i.i.d. initialization of u , g , and f , we have that $\kappa_{xx'}^\epsilon = \kappa_{xx'}^u + \kappa_{xx'}^g = 2\kappa_{xx'}^u = 2\kappa_{xx'}$ and accordingly $\frac{1}{2}\Sigma_{xx}^\epsilon(\vartheta_\infty, \psi_0) = \Sigma_{xx}^f(\theta_\infty)$, completing the proof. $\square$

B.2 POSTERIOR EQUIVALENCE

This section contains proofs for results pertaining to the equivalence of self-predictive errors of ‘‘Bayesian RND’’ and the variance of Bayesian posterior predictive distributions of neural networks in the infinite width limit.

B.2.1 Proof of Proposition 4.1

We restate Proposition 4.1 for convenience.

Proposition 4.1. (Bayesian RND target function) Under the conditions of Theorem 3.1, let $u(x; \vartheta_0)$ and $g(x; \psi_0)$ be neural networks of L layers with parameters $\vartheta_0, \psi_0 \sim \mathcal{N}(0, I)$ i.i.d. Moreover, let $\psi_0^L = \{w^L, b^L\}$ denote the last-layer parameters of ψ_0 and $\psi_0^{\leq L-1}$ the parameters of all preceding layers. Suppose the target function $\tilde{g}(x; \vartheta_0, \psi_0)$ is given by

$$\tilde{g}(x; \vartheta_0, \psi_0) = \nabla_{\vartheta_0} u(x; \vartheta_0)^\top \psi_0^*,$$

where $\psi_0^* = \{\psi_0^{\leq L-1}, 0_{\dim(\psi_0^L)}\}$ is a copy of ψ_0 with its last-layer weights set to 0. In the infinite width limit $n \rightarrow \infty$, $\tilde{g}(x; \vartheta_0, \psi_0)$ distributes by construction as $\tilde{g}(x; \vartheta_0, \psi_0) \sim \mathcal{GP}(0, \kappa_{xx'}^{\tilde{g}})$ where $\kappa_{xx'}^{\tilde{g}} = \Theta_{xx'}^{\leq L-1}$.

Proof. The proof will show that in the limit $n \rightarrow \infty$ the function $\tilde{g}(x; \vartheta_0, \psi_0)$ converges to a GP $\tilde{g}(x; \vartheta_0, \psi_0) \sim \mathcal{GP}(0, \Theta_{xx'}^{\leq L-1})$ by Lévy's continuity theorem, which we recall informally below.

Theorem B.4. (Lévy's continuity theorem) Let $\{Z_n\}_{n=1}^\infty$ be a sequence of $\mathbb{R}^n$ -valued random variables. Their characteristic functions $\varphi_{Z_n}(t)$ for some $t \in \mathbb{R}^n$ are given by

$$\varphi_{Z_n}(t) = \mathbb{E}[e^{it^\top Z_n}], \quad (71)$$

where i is the imaginary unit. If in the limit $n \rightarrow \infty$ the sequence of characteristic functions converges pointwise to a function

$$\varphi_{Z_n}(t) \rightarrow \varphi(t) \quad \forall t \in \mathbb{R}^n, \quad (72)$$

then Z_n converges in distribution to a random variable Z

$$Z_n \xrightarrow{D} Z, \quad (73)$$

whose characteristic function is $\varphi_Z(t) = \varphi(t)$

Rigorous proof can be found for example in Durrett [2019].

We begin by rewriting the function $\tilde{g}(x; \vartheta_0, \psi_0)$ as a linear model with

$$\tilde{g}(x; \vartheta_0, \psi_0) = \nabla_{\vartheta} u(x; \vartheta_0)^\top \psi_0^* \quad (74)$$

$$= \nabla_{\vartheta \leq L-1} u(x; \vartheta_0)^\top \psi_0^{\leq L-1}. \quad (75)$$

Since $\psi_0^{\leq L-1}$ is an independent draw from ϑ_0 by assumption, $\tilde{g}(x; \vartheta_0, \psi_0)$ is a random affine transform of the Gaussian vector $\psi_0^{\leq L-1}$. For more precise treatment of the distribution of $\tilde{g}(x; \vartheta_0, \psi_0)$, we write $\tilde{G}(\mathcal{X}_T)$ to denote the random variable corresponding to the function evaluations of $\tilde{g}$ on a test set $\mathcal{X}_T$. Conditioned on ϑ_0 (i.e., fixing the affine transform), we thus have that $\tilde{G}(\mathcal{X}_T)|\vartheta_0 \sim \mathcal{GP}(0, \Theta_{0, \mathcal{X}_T \mathcal{X}_T}^{\leq L-1})$, where $\Theta_{0, \mathcal{X}_T \mathcal{X}_T}^{\leq L-1} = \nabla_{\vartheta \leq L-1} u(\mathcal{X}_T; \vartheta_0)^\top \nabla_{\vartheta \leq L-1} u(\mathcal{X}_T; \vartheta_0)$ is the empirical NTK matrix of u . Note that this statement holds irrespective of the network width n .

Next, we show that the unconditional law of $\tilde{G}(\mathcal{X}_T)$, too, tends to a GP in the limit $n \rightarrow \infty$. To this end, we examine the distribution of the unconditioned random vector $\tilde{G}(\mathcal{X}_T)$ through its characteristic function

$$\varphi_{\tilde{G}(\mathcal{X}_T)}(t) = \mathbb{E}[e^{it^\top \tilde{G}(\mathcal{X}_T)}]. \quad (76)$$

This characteristic function $\varphi_{\tilde{G}(\mathcal{X}_T)}(t)$ uniquely defines the distribution of $\tilde{G}(\mathcal{X}_T)$ [Durrett, 2019]. By the law of total expectation, the characteristic function of the unconditional variable $\tilde{G}(\mathcal{X}_T)$ can then be written as

$$\varphi_{\tilde{G}(\mathcal{X}_T)}(t) = \mathbb{E}_{\vartheta_0} [\mathbb{E}[e^{it^\top \tilde{G}(\mathcal{X}_T)} | \vartheta_0]]. \quad (77)$$

As stated above, the conditional distribution of $\tilde{G}(\mathcal{X}_T)|\vartheta_0$ is a zero-mean Gaussian with the empirical covariance $\Theta_{0, \mathcal{X}_T \mathcal{X}_T}^{\leq L-1}$, to which we can show the conditional characteristic function is given by [Durrett, 2019]

$$\mathbb{E}[e^{it^\top \tilde{G}(\mathcal{X}_T)} | \vartheta_0] = e^{-\frac{1}{2} t^\top \Theta_{0, \mathcal{X}_T \mathcal{X}_T}^{\leq L-1} t}. \quad (78)$$

Plugging this back into Eq. 77 gives

$$\varphi_{\tilde{G}(\mathcal{X}_T)}(t) = \mathbb{E}_{\vartheta_0}[e^{-\frac{1}{2}t^\top \Theta_{0,\mathcal{X}_T\mathcal{X}_T}^{\leq L-1} t}]. \quad (79)$$

We now use the known result by Jacot et al. [2018] that, as $n \rightarrow \infty$ we have that $\Theta_{0,\mathcal{X}_T\mathcal{X}_T} \rightarrow \Theta_{\mathcal{X}_T\mathcal{X}_T}$ in probability and accordingly $\Theta_{0,\mathcal{X}_T\mathcal{X}_T}^{\leq L-1} \rightarrow \Theta_{\mathcal{X}_T\mathcal{X}_T}^{\leq L-1}$ converges to a deterministic kernel matrix. Moreover, since the Gram matrix $\Theta_{0,\mathcal{X}_T\mathcal{X}_T}^{\leq L-1}$ is positive semidefinite in general, the term $e^{-\frac{1}{2}t^\top \Theta_{0,\mathcal{X}_T\mathcal{X}_T}^{\leq L-1} t}$ is bounded and continuous. By bounded convergence [Durrett, 2019], we can then conclude that we also have convergence of the characteristic function through

$$\lim_{n \rightarrow \infty} \varphi_{\tilde{G}(\mathcal{X}_T)}(t) = \lim_{n \rightarrow \infty} \mathbb{E}_{\vartheta_0}[e^{-\frac{1}{2}t^\top \Theta_{0,\mathcal{X}_T\mathcal{X}_T}^{\leq L-1} t}] \quad (80)$$

$$= e^{-\frac{1}{2}t^\top \Theta_{\mathcal{X}_T\mathcal{X}_T}^{\leq L-1} t}. \quad (81)$$

As stated earlier, for a Gaussian random vector Z with $Z \sim \mathcal{GP}(0, \Theta_{\mathcal{X}_T\mathcal{X}_T}^{\leq L-1})$ its characteristic function is given by $e^{-\frac{1}{2}t^\top \Theta_{\mathcal{X}_T\mathcal{X}_T}^{\leq L-1} t}$. Invoking Lévy's continuity theorem, the pointwise convergence of $\varphi_{\tilde{G}(\mathcal{X}_T)}(t)$ to this exact limit $\varphi_{\tilde{G}(\mathcal{X}_T)}(t) \rightarrow e^{-\frac{1}{2}t^\top \Theta_{\mathcal{X}_T\mathcal{X}_T}^{\leq L-1} t}$ then implies convergence in distribution of $\tilde{G}(\mathcal{X}_T) \xrightarrow{D} Z$ and we can thus conclude $\tilde{g}(x; \vartheta_0, \psi_0) \sim \mathcal{GP}(0, \Theta_{xx'}^{\leq L-1})$. $\square$

B.2.2 Proof of Theorem 4.2

We restate Proposition 4.1 for convenience.

Theorem 4.2. (*Distribution of Bayesian RND errors*) Under the conditions of Theorem 3.1, let $u(x; \vartheta_\infty)$ be a converged predictor network trained on data $\mathcal{X}$ with labels from the fixed target function $\tilde{g}(\mathcal{X}; \vartheta_0, \psi_0)$ as defined in Proposition 4.1. Let parameters ϑ_0, ψ_0 be drawn i.i.d. $\vartheta_0, \psi_0 \sim \mathcal{N}(0, I)$. The converged Bayesian RND error $\epsilon^b(\mathcal{X}_T; \vartheta_\infty, \vartheta_0, \psi_0) = u(\mathcal{X}_T; \vartheta_\infty) - \tilde{g}(\mathcal{X}_T; \vartheta_0, \psi_0)$ on a test set $\mathcal{X}_T$ is Gaussian with zero mean and covariance

$$\Sigma_{\mathcal{X}_T\mathcal{X}_T}^{\epsilon^b}(\vartheta_\infty, \vartheta_0, \psi_0) = \Theta_{\mathcal{X}_T\mathcal{X}_T} - \Theta_{\mathcal{X}_T\mathcal{X}} \Theta_{\mathcal{X}\mathcal{X}}^{-1} \Theta_{\mathcal{X}\mathcal{X}_T},$$

and thus recovers the covariance of the exact Bayesian posterior predictive distribution of an infinitely wide neural network with the corresponding NTK $\Theta_{xx'}$.

Proof. The result follows from the independence of the two GP of interest in the limit $n \rightarrow \infty$. First, this is $\tilde{g}(x; \vartheta_0, \psi_0) \sim \mathcal{GP}(0, \Theta_{xx'}^{\leq L-1})$ and second, $u(x; \vartheta_0) \sim \mathcal{GP}(0, \Theta_{xx'}^L)$. In the following, we will show that the two GPs are in the limit $n \rightarrow \infty$ independent processes such that Eq. 14 applies.

We first write for any two points x, x' the covariance

$$\text{Cov}[\tilde{g}(x; \vartheta_0, \psi_0), u(x'; \vartheta_0)] = \mathbb{E}[\tilde{g}(x; \vartheta_0, \psi_0)u(x'; \vartheta_0)]. \quad (82)$$

As ψ_0 is drawn independently of ϑ_0 , the conditional expectation can be written as

$$\mathbb{E}[\tilde{g}(x; \vartheta_0, \psi_0)u(x'; \vartheta_0)|\vartheta_0] = u(x'; \vartheta_0)\mathbb{E}[\tilde{g}(x; \vartheta_0, \psi_0)|\vartheta_0] \quad (83)$$

$$= u(x'; \vartheta_0)\mathbb{E}[\nabla_{\vartheta \leq L-1} u(x; \vartheta_0)^\top \psi_0^{\leq L-1}|\vartheta_0] \quad (84)$$

$$= u(x'; \vartheta_0) \cdot 0, \quad (85)$$

and by the law of total expectation

$$\mathbb{E}[\tilde{g}(x; \vartheta_0, \psi_0)u(x'; \vartheta_0)] = \mathbb{E}_{\vartheta_0}[\mathbb{E}[\tilde{g}(x; \vartheta_0, \psi_0)u(x'; \vartheta_0)|\vartheta_0]] \quad (86)$$

$$= 0. \quad (87)$$

We conclude that the two GP $\tilde{g}(x; \vartheta_0, \psi_0) \sim \mathcal{GP}(0, \Theta_{xx'}^{\leq L-1})$ and $u(x; \vartheta_0) \sim \mathcal{GP}(0, \Theta_{xx'}^L)$ are mutually independent such that the initialization kernel $\kappa_{xx'}^{\epsilon^b}$ is given as

$$\kappa_{xx'}^{\epsilon^b} = \Theta_{xx'}. \quad (88)$$

This is because $\Theta_{xx'} = \Theta_{xx'}^L + \Theta_{xx'}^{\leq L-1}$ and $\kappa_{xx'}^{\tilde{g}} = \Theta_{xx'}^{\leq L-1}$, $\kappa_{xx'}^u = \Theta_{xx'}^L$ are mutually independent. $\square$

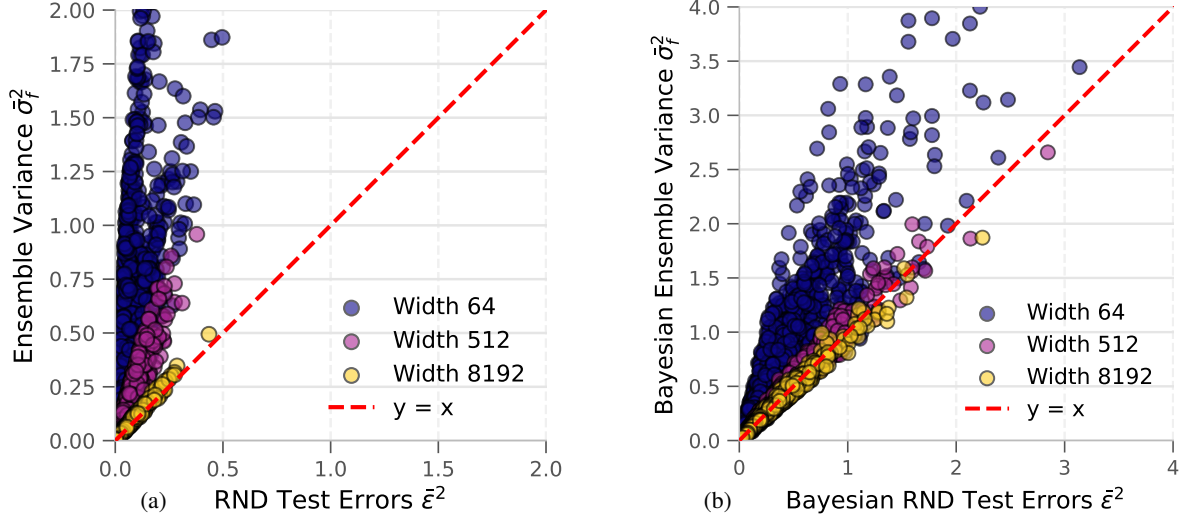


Figure 4: (a) Scatter plot of test-set errors between predictive variances of ensembles and self-predictive errors of RND. As width increases, errors become more correlated and correctly calibrated in scale. (b) Likewise, for *Bayesian* ensembles and *Bayesian* RND.

C ADDITIONAL EXPERIMENTAL RESULTS

We report additional experimental details and evaluations. As outlined in the main text, we use two-layer fully connected neural networks with SiLU activations and NTK parametrization. All weights and biases are initialized as $\theta \sim \mathcal{N}(0, I)$. We use an ensemble of 512 models and a single multiheaded RND network with 512 heads. A synthetic dataset is generated with $N = 10$ train and $\tilde{N} = 5000$ test samples from an isotropic Gaussian $x \sim \mathcal{N}(0, I_3)$. We label training samples with a synthetic target function

$$y(x) = x^0 + x^1 + x^2 - 2 \prod_{i=1}^3 x^i, \quad (89)$$

where x^i denotes the i -th component of vector x . All models are trained according to the algorithms outlined in the main text. For this, we use full-batch gradient descent with a learning rate of 0.1 for all models. Fig. 4 shows additional results of the same experiment, in which we plot individual test-set ensemble variances against RND errors. As the network width increases, ensemble variances and self-predictive RND errors become more correlated and well-calibrated in scale.

Two-Moons: Deviation vs. Width. We repeat the experiment underlying Fig. 2 on the two-moons dataset described in the main text, using the same network architecture, ensemble size, and number of RND heads. Training data is sampled with a noise parameter of 0.1, while test points used to evaluate the discrepancy between RND/ensembles and their Bayesian counterparts are sampled from a wider two-moons distribution with a noise parameter of 0.3, such that test points extend beyond the immediate training support; reported discrepancies are averaged over all test points. Fig. 5 shows that the same trend of vanishing discrepancy between RND/ensembles and their Bayesian counterparts, observed on the synthetic Gaussian dataset, persists on two-moons as network width increases.

Ablations: Initialization and Minibatching. We additionally test the robustness of our equivalences to two departures from the idealized setting assumed in our theoretical analysis: (i) the choice of initialization scheme, and (ii) the use of minibatched stochastic gradient descent in place of full-batch gradient descent. For (i), we compare standard Gaussian initialization against truncated Gaussian, uniform, and orthogonal initialization schemes, all rescaled to unit variance: the truncated Gaussian is rescaled by a fixed factor of $1/0.8796$, the uniform distribution uses bounds $\pm\sqrt{3}$, and orthogonal initialization is rescaled by $\sqrt{\max(d_{\text{in}}, d_{\text{out}})}$, where d_{in} and d_{out} denote the input and output dimensions of the respective layer. For (ii), we replace full-batch gradient descent with minibatched SGD using a minibatch size of 32. Both ablations otherwise use the same two-moons training/test setup described above. Fig. 6 reports the resulting test-set divergence between RND/ensembles and their Bayesian counterparts across widths for both ablations. Across all initialization schemes and under minibatched optimization, we observe the same diminishing discrepancy with increasing width as in our main

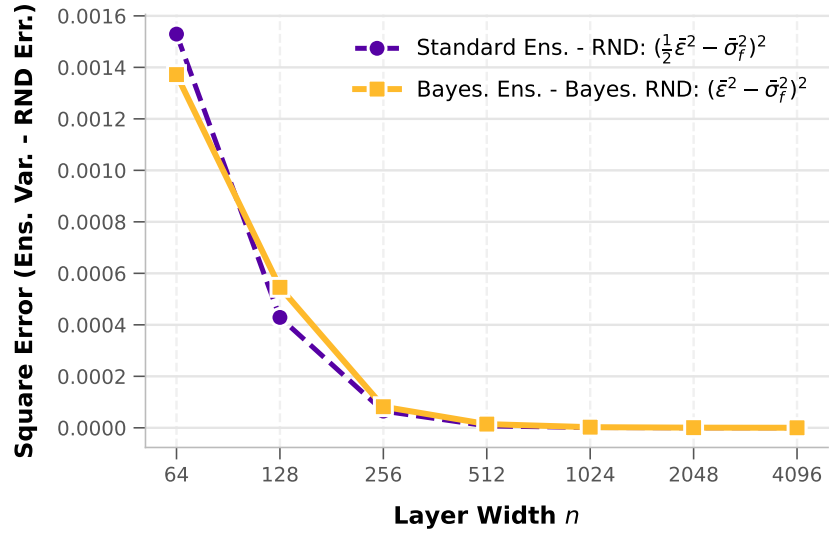


Figure 5: Test-set errors between predictive variances of (Bayesian) ensembles and self-predictive errors of (Bayesian) RND vanish with large layer widths on the two-moons dataset, mirroring the trend observed in Fig. 2.

experiments, indicating that the established equivalences can hold beyond the strict conditions assumed in the theoretical setting.

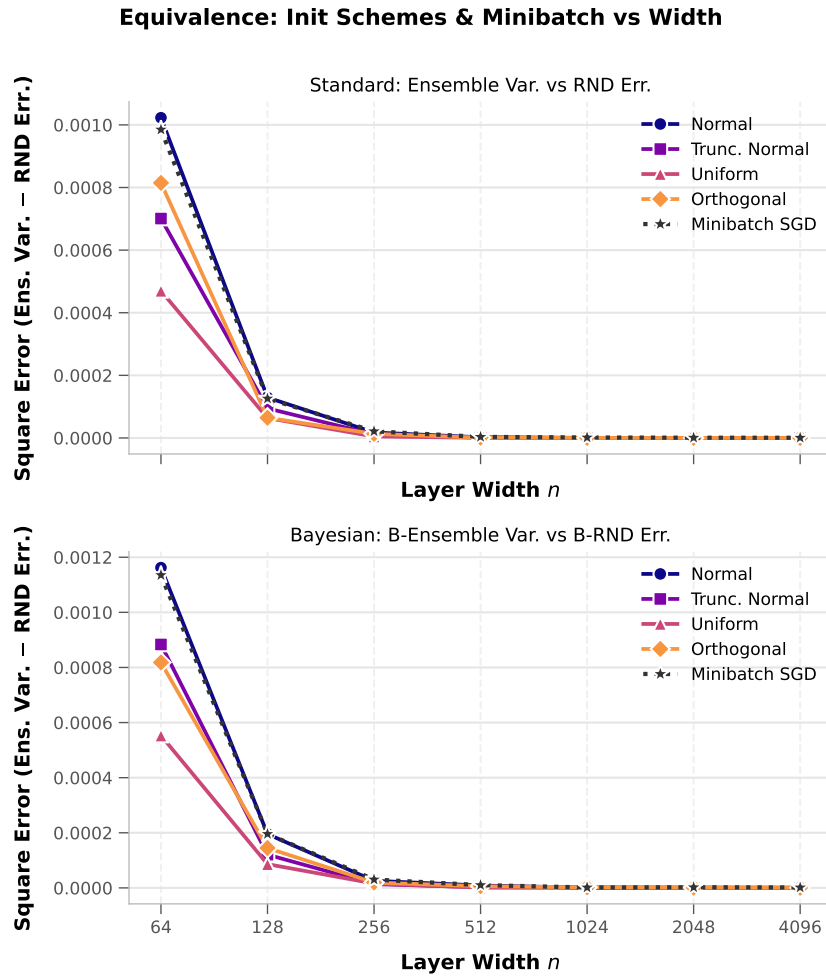


Figure 6: Test-set divergence between RND/ensembles and their Bayesian counterparts across widths, under alternative initialization schemes (truncated Gaussian, uniform, orthogonal) and under minibatched SGD with batch size 32. The same width-dependent vanishing discrepancy persists across all variations.